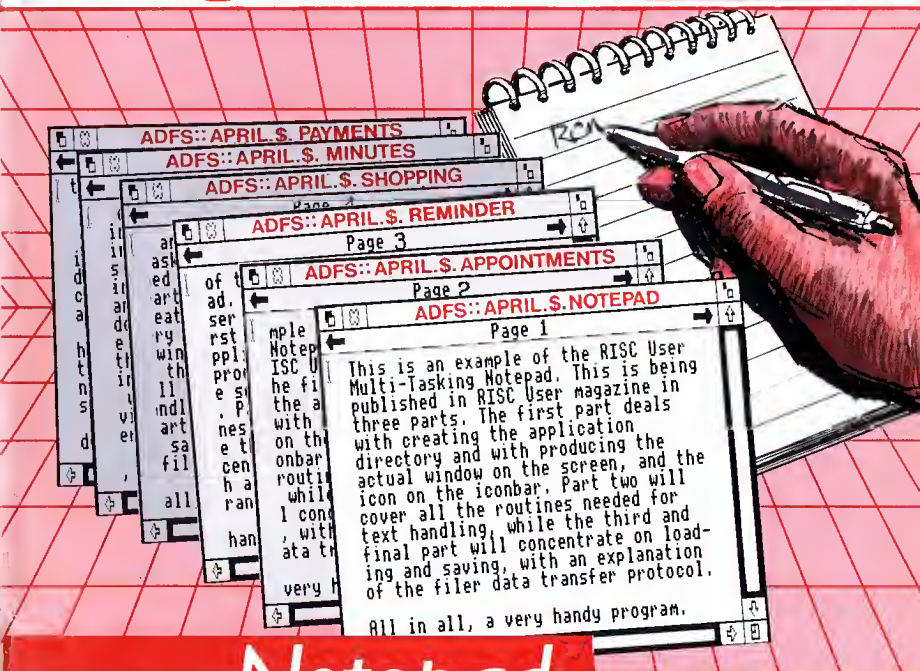
Volume 2
Issue 5

April
1989

Price £1.20

RISC USER



Notepad

THE MAGAZINE AND SUPPORT GROUP
EXCLUSIVELY FOR USERS OF THE ARCHIMEDES

Volume 2 Issue 5
April 1989

RISC USER

CONTENTS

FEATURES

News	4
Acorn's Ray Tracer	15
The RISC OS Hourglass	21
ARC Procedure Library	29
The RISC OS RAM Filing System	31
Using Basic's Token Table	33
ArcEdit: The RISC OS Text Editor	35
Archimedes Visuals Real Time Image Wrapping	38
Postbag	51
Hints & Tips	52

UTILITIES AND APPLICATIONS

The RISC User Notepad	9
Basic Program Listing Utility	26
Using the RISC User Toolbox with RISC OS	49

REVIEWS

UNIX for the ARM	5
Art Nouveau	23
Computer Concept's Scanner	42
Archimedes Operating System (A User's Guide)	45

RISC User is published by BEEBUG Ltd.
Co-Editors: Mike Williams, Dr Lee Calcraft
Assistant Editor: Kristina Lucas
Technical Editor: David Spencer
Technical Assistant: Glynn Clements
Production assistant: Sheila Stoneman
Advertising: Sarah Shrive
Subscriptions: Mandy Milleham
BEEBUG Ltd (c) 1989

All rights reserved. No part of this publication may be reproduced without prior written permission of the Publisher. The Publisher cannot accept any responsibility whatsoever for errors in articles, progrems, or advertisements published. The opinions expressed on the pages of this journal are those of the authors and do not necessarily represent those of the Publisher, BEEBUG Limited.

BEEBUG, Dolphin Place, Holywell Hill, St.Albans,
Herts AL1 1EX. Tel. St.Albans (0727) 40303

Printed by Arlon House

EDITORIAL

RISC OS AGAIN

We make no apologies for raising the issue of RISC OS once again in our editorial. All reports from Acorn indicate that the company has initial supplies of the RISC OS upgrade already in stock, and that kits will start to go out to users from early April. We also expect further announcements from Acorn about the Archimedes range at the same time.

We hope we have already managed to convey to you some of the enthusiasm and excitement which we feel about RISC OS and its associated software. It is a very significant step forward for the Archimedes, and will do more than anything to enhance its reputation in the market-place.

This is likely to be the last issue of RISC User in which we will separately identify RISC OS items from the rest, as in the future RISC OS will be the standard to which we shall be working. However, RISC OS is a development of the existing Arthur 1.2, and any programs which we publish, unless specifically using any of the new features of RISC OS, will continue to work happily on existing systems.

We feel that it is important for the Archimedes world that all users upgrade to RISC OS at the earliest possible opportunity, and we hope that RISC User readers will be in the forefront of this movement. The cost (£33.35 plus £3 p&p if ordered from BEEBUG) is very reasonable for what you will receive (two new manuals, three discs and a set of four chips), but more importantly it will transform your Archimedes into a superlative machine.

This month's telesoftware password is ***Hebrides.***
(See BEEBUG pages on Micronet)

RISC OS IS AROUND

If you placed an advance order with BEEBUG for your copy of RISC OS, then the chances are that you will have it by the time you read this. It is known that Acorn are currently building up stocks of the new operating system, and at the time of writing (end of February) it is thought that they will be released around the middle of March. The upgrade kit consists of four ROMs, an installation guide, a new Welcome guide, a new User Guide, and three discs. Two of these are 'Welcome' discs, while the third contains application directories for use with commercial software. The complete system is packed in a sturdy foam-lined white cardboard sleeve. If you haven't ordered RISC OS, then you can still do so from BEEBUG. The price to members is £36.35 including VAT and postage.

SHOWING OFF

It had been thought that users of Acorn equipment would have been deprived of any specialist shows this year, because Database Publications decided not to hold their traditional May and Novmember shows. However, Redwood Publishing, organisers of the Acorn User show, which was last year called off, were quick to fill the gap. They will be holding an Acorn User show at Alexandra Palace from the 21st to 23rd of July. This is a day shorter than previous AU shows, and at a venue rather more out of the way than the Barbican.

BE KIND TO YOUR KEYBOARD

There is nothing better at ruining a keyboard than a nice hot cup of coffee. Recognising this, Kador has produced a range of *Seal 'n' Type* keyboard covers that protect the keyboard in every-day use. The covers are made of very thin, but highly durable, transparent PVC, which protects the keys while retaining the 'feel' of the keyboard. The *Seal 'n' Type* cover for the Archimedes costs £11.95 inclusive, and is available from Kador, Unit 4, Pontcynon Industrial Estate, Abercynon, Mid Glamorgan CF45 4EP, tel. (0443) 740281. Kador also produce a range of *Peace 'n' Quiet* noise excluders for printers.

A PREMIER PROGRAM

Premier, a recently released text processor from Circle Software, is claimed to be a new concept in word processors. Documents can include so-called *Dynamic Text* which is text that can be updated at a later date without changing the actual file. For example, a table of values can be incorporated into a document, with the actual values being saved in a separate file. This means the table can be changed just by altering the values file. Premier costs £50 (inc. VAT), which includes any updates released this year. A demo disc is available for £5. More details from Circle Software, 33 Restrop View, Purton, Swindon, Wiltshire SN5 9DG.

UNIX DEAL FOR SCHOOLS

Acorn has launched a special drive to encourage schools to buy the R140 UNIX system reviewed in this issue. Local Education authorities are being given the chance to purchase the R140 complete with a colour monitor and two school administration packages (SIMS and SCRIPT) for £2695 (ex. VAT). This represents a saving of nearly £1500 over the normal price.

BEAT THE BUDGET

Instead of holding a post-mortem on the budget speech this year, you can try and do better yourself with a package from Topologika. *Yes Chancellor!* is a complete simulation of the country's economy, including inflation, stock market fluctuations, and exchange rates, as well as unforeseen events such as strikes and election fever. *Yes Chancellor!* was originally released a number of years ago for the BBC model B, but the new version is for an Archimedes equipped with the PC Emulator. *Yes Chancellor!* costs £18 (inc. VAT) from Topologika, PO Box 39, Stilton, Peterborough PE7 3RL, tel. (0487) 831153.

GATE ARRAY DESIGNER

Silicon Vision has launched a package to simulate the design of gate arrays - complex custom-made chips which form the heart of many computers. Real chip design systems cost many thousands of pounds, and are therefore impractical to use for teaching purposes. However, Silicon Vision's system sells for just £89.95 (inc. VAT). More details can be obtained from Silicon Vision, Signal House, Lyon Road, Harrow, Middlesex HA1 2AG, tel. 01-422 2274.

RU

UNIX FOR THE ARM

David Spencer investigates Acorn's entry into the UNIX marketplace.

Acorn has for a long time been trying to penetrate the business sector. However, the company's success has not been notable until now (remember the Acorn Business Machine and the Cambridge Workstation?) This time Acorn is attacking the growing market of super-micros and small mini-computers. The machine with which they are doing this is the R140 UNIX Workstation.

The Acorn R140 Unix system

THE HISTORY

Before describing the R140 I will give a brief history of UNIX. It all started many years ago when people first realised that computers were to play a major part in the everyday life of the future. A pioneering multi-user operating system called MULTICS was developed for the new breed of practical computers. Unfortunately, it was too advanced for the hardware of the time and was not widely used. Several years later a team working for the Bell Corporation (now AT&T) converted MULTICS to a single-user system which they called UNIX (UNI for one as opposed to MULTI for many). Unlike earlier operating systems which were hand coded in machine code, UNIX was (and still is) written in a new high-level language developed specially for the job. This language is none other than C. The use of a high level language means that UNIX can easily be ported to any computer, regardless of the processor it uses. Contrast this with something like RISC OS which is only for Archimedes type computers, or even MS-DOS which while included on many different computers will only run with an 8086 processor, or similar.

The major drawback of UNIX is the size of system needed to run it. A typical UNIX-based computer will need several megabytes of RAM and tens of megabytes of disc storage. This has tended to restrict UNIX to mini and mainframe systems in the past. Now though, the increase in power of micros has made UNIX a viable proposition for small systems.

THE R140

At first sight you might think that the R140 is an ordinary Archimedes 440, the only visible difference being the stickers on the front panel and the keyboard. Indeed, the similarity continues inside. The most obvious change is the inclusion of a larger hard disc drive (about 56Mb). This extra storage is mostly used by the UNIX system software. The R140 also uses a faster version of the MEMC chip (the device that controls memory access) which will allow for future speed improvements, although it offers no advantage at the present time. The only other change is a slightly redesigned podule backplane which avoids possible problems that were discovered while developing the Ethernet podule described later.

Seeing the similarity between the 440 and the R140 you might expect Acorn to offer an upgrade path. However this will not be the case, more I suspect for marketing strategy than technical reasons.

STARTING UNIX

When you first turn on the R140 you are presented with the standard RISC OS Desktop, the only difference being a UNIX icon that appears on the iconbar along with the resident applications. At this point the computer can be used as a normal RISC OS-based Archimedes. A partition is set up on the hard disc making 3Mb available to this mode of operation.

Clicking on the UNIX icon brings up a warning that RISC OS is about to be shut-down and UNIX started. At this stage you can use the

mouse to perform a number of diagnostic functions, or you can acknowledge the message which will cause UNIX to start up.

The first thing to strike you when UNIX is started is the total loss of colour. The friendly RISC OS Desktop display gives way to a few lines of black and white text asking you to log on. UNIX insists that you log on as a known user before giving access to any part of the system. It is actually possible to log on more than once, and it is even possible to have two users logged on simultaneously, one using the R140, and the other using another computer as a terminal, this being connected to the R140 via the serial port or the network.

UNIX COMMANDS

As UNIX has its basis in early mainframe operating systems, many of the commands offered are quite obscure. For example, the command to list the files in a directory (equivalent to *CAT) is 'ls'. The operation of most commands can be modified by a number of single letter qualifiers placed after the command name.

To make the use of UNIX commands easier, the system features an entire on-line user guide. This is invoked by typing 'man <command>', where command is the name of the command to be described. This displays full documentation for the specified command.

WINDOWS AND THE DESKTOP

Supplied with the R140 is a window system and desktop developed by IXI Limited. This is similar to the RISC OS window system, and allows the same tasks to be performed. One interesting feature is that the layout of the control areas of a window can be changed. For example, you can move the scroll-bars to the opposite sides.

The IXI Desktop manager is again much like that of RISC OS. However, instead of having an iconbar, icons can be placed anywhere on the screen. Furthermore, at any point, a chosen window can be reduced to an icon, and then reopened when needed.

FILING SYSTEMS AND NETWORKS

The floppy disc drive of the R140 can read several different formats, including those used by the majority of other UNIX systems. However, most UNIX systems will use a network, and an Ethernet interface is available for this, either with the R140, or as a later upgrade. Ethernet is a system providing similar features to Acorn's Econet, but much faster. It is a standard system used by most UNIX installations. From a user's point of view, the R140 supports Sun's NFS system which extends the hierarchical directory structure so that each computer's entire disc contents is contained as a sub-directory in a 'Super-directory'. This means that any computer on the network can read a file from any other just by specifying the computer to use in the pathname of the file.

DOCUMENTATION

In terms of sheer volume, the documentation supplied with the R140 is impressive. There is a total of six manuals, a lengthy release note and a licensing agreement, all supplied in a box separate from the main hardware. The first three manuals are the standard RISC OS Welcome and User Guides as supplied with RISC OS, and the new BBC Basic Guide to be supplied with 400 series machines.

The other three manuals are dedicated to the UNIX system. The first of these is a Welcome Guide which explains about the machine and UNIX, and how to get it all set up and started. The UNIX User guide is much bigger and more comprehensive than the introductory work. This manual covers the main features that an average UNIX user will need while using the R140. The final manual explains the IXI Window and XDesktop system, and includes the actual IXI software which must be installed from floppy initially. What none of the manuals cover at all are the vast number of commands offered by UNIX, or the use of the supplied C compiler. This may sound like a major omission, but remember that UNIX is totally standard, and therefore any UNIX manual or text book will do. As most users will not need this information Acorn decided to recommend a suitable manual rather than supplying one.

HOW DOES IT COMPARE?

With UNIX being a fairly stringent standard, the one thing that does set machines apart is their performance. Table 1 gives the average result of a number of standard UNIX benchmark timings for several UNIX systems. (The lower the value, the faster the speed). The approximate price is also given for reference.

Computer:	R140	VAX 11/780	SUN 3/50	SUN 386i/250	APPLE MAC II
Benchmark:	1.6	3.3	2.2	0.9	1.7
Price:	£4500	£30000	£5850	£12105	£7500

Only the 80386-based top of the range Sun is faster than the R140. Indeed, the R140 is twice as fast as the £30000 VAX mainframe.

APPLICATIONS

I have deliberately not mentioned application software for the R140. This is because no software is supplied. However, the UNIX standard ensures that, in theory, all UNIX software can be ported to the R140. Needless to say, things are not quite so easy in practice, although the current R140 software catalogue does include packages to perform most common tasks, such as word processing and database management.

PRICE AND AVAILABILITY

The standard R140 costs £3500 (ex. VAT), or £3949 with the Ethernet podule. These prices do not include a monitor. The Ethernet podule is also available separately for £449 (ex. VAT), and a hardware floating point co-processor and SCSI interfaces are promised in the future. The latter will allow additional hard disc drives to be connected, along with tape streamers etc. Acorn has decided to market the R140 through UNIX dealers, rather than the company's existing dealer network. They say that this is so that the machine can be given the support it needs, and so that customers can see it in comparison to other systems.

RU

"FIRST"

ADVANCED STATISTICAL SYSTEM

For Acorn Archimedes; BBC Models B, B+ and Master; "Industry Standard" machines

A powerful and practical tool for Research, Industry, Business, and Teaching

Integrated, interactive, robust, fast, accurate, modular. Designed to facilitate data critique.
Many data entry options. Full data management and formatted display. Extensive data transforms.
Handles missing variables and data subsets. Scatterplots, regression plots, function plots.
Univariate statistics. Paired and unpaired t tests. Chisquare tests. Nonparametric methods.
Unrivalled REGRESSION facilities eg weighted, through the origin, standardised. Full statistical information.
Residuals, fitted and predicted values. Much more. Correlations. ANOVA. Contour plots of regressions.
Regression DIAGNOSTICS - VIFs, influential points etc. Automatic warnings.
ROBUST regression - many influence functions. Powerful NONLINEAR least squares
Time Series, eg moving averages, exponential smoothing, causal models.
One- Two- and Three-way ANOVA subsystems. Distributions generator. Histograms. Linear Calibration.
Cluster analysis. Many other powerful analytical and descriptive features. Full utilities.

Price £120-£180 (machine and version dependent). Special RISC USER discount available.

There's for too much to describe here. Get full information now from:

Serious Statistical Software, Lynwood, Benty Heath Lane, Willaston, South Wirral L64 1SD Tel. 051 327 4268

Attention RISCOS Users!!

A New Communications Package from BBC Soft

Taking full advantage of RISCOS, ArcComm is an exciting, brand-new communications package for the Archimedes that you can't afford to be without!

ArcComm's comprehensive design allows access to all three European videotex standards (Prestel, Télétel, and Bildshirmtext), together with VT52, VT102, and ANSI terminals.

It's wonderfully simple to use! All screen displays are clear and uncluttered, with pop-up menus and dialogue boxes, making control as easy as possible.

ArcComm supports Hayes, DTI, and manual modems, with details of how to write drivers for other modems in the User Guide provided. You will need RISCOS, and at least 1Mbyte of memory to use ArcComm.

To order ArcComm, simply write to the address shown, enclosing a cheque. ArcComm is £29.95 inc VAT, plus £1.50 p&p per order.

BBC Soft,
PO Box 234,
WETHERBY,
West Yorkshire,
LS23 7EU.

THE RISC USER NOTEPAD

David Spencer presents a genuine RISC OS multi-tasking application.

Over the next three issues, I will be showing how to write a fairly simple, but comprehensive, multi-tasking application for RISC OS. The program will be a notepad utility, allowing short notes and memos to be 'jotted down', and either saved for later use or printed out. This application was chosen both because it has some practical use, and is also basic enough not to result in a massively long program. Although the purpose of these articles is to demonstrate the development of a RISC OS multi-tasking application, it is not feasible to give a complete line by line account of the program because of the complexity of the overall system. To understand the workings of the program, a basic knowledge of the WIMP system is necessary.

THE APPLICATION DIRECTORY

The first step in writing our application is to create an application directory as explained last month (page 20). This directory will have the name '!NotePad', and will contain five files:

```
!Boot
!Run
!Sprites
!RunImage
Sprites
```

The purpose of the first three was explained in the article last month. The fourth one, '!RunImage', is the application program itself, while 'Sprites' will contain various sprites used by the program.

You should start by creating the directory anywhere on your working disc using *CDIR !NotePad. Secondly, the !Boot and !Run obey files should be entered, and saved in the application directory. The contents of these files is given in listings 1 and 2 respectively. The comment lines (those starting with '|' can be omitted if you wish, but for clarity I suggest that at least the filename appears somewhere in the file. The command files can be entered using a text editor such as *Twin* or RISC OS's *ArcEdit* (see article in this issue), or using *BUILD. However, in the case of *BUILD, all occurrences of '|' and '<' in the text must be preceded by a '|' (giving '||' and '|<') when the text is entered. This prevents *BUILD from trying to interpret the lines as they are entered. In all cases, the file type of both files should be set to 'Obey'. If you are using *ArcEdit*, this will be done automatically provided you start by selecting the 'Create Obey File' option from the iconbar menu. If the files are entered any other way, the commands:

```
*SETTYPE $.!NotePad.!Boot Obey
*SETTYPE $.!NotePad.!Run Obey
```

will be needed to set the correct filetype.

The comments in the two obey files should give you an idea of the functions they perform, especially if you followed last month's article on application directories. The !Boot file sets up a name for a chosen filetype and then an alias to specify what happens when a Notepad is run. File type &FEE is used because this was previously allocated to the Arthur 1.20 Notepad. The final action of !Boot is to load in the sprites used to represent the application, and its data files, in a directory viewer.

The !Run file starts by repeating all the actions of !Boot, because in some cases it is possible that !Boot is never executed (for example, if two applications have the same name). It then sets the system variable 'NotePad$Dir' to the pathname of the application directory and ensures that 32K of memory is available. This latter function is achieved using the command *WimpSlot which informs the task manager of the amount of memory needed by this application. Finally, !Run actually runs the Basic program !RunImage which is the application itself.

SPRITE FILES

The next stage is to create the two sprite files needed by the application. Specifying sprite definitions in a magazine article is particularly awkward, and we have decided to adopt the approach of listing 3. This reads the definitions from a string of DATA statements, and creates the two sprite files. If you have any alternative suggestions for specifying sprite definitions in the future then please let us know. One point to watch out for is that this listing is formatted to 43 columns for readability, rather than our usual 40.

The file '!Sprites' contains sprites called '!notepad' and 'file_fee'. The first of these, which has the same name as the application directory, is used to display an icon for the application in the filer, and we will also use it to represent the application on the iconbar. The second sprite is used for data files that are recognised by our application. (Any file of filetype &FEE.)

The second sprite file, 'Sprites' contains sprites used solely by this particular application. There are two sprites called 'leftarrow' and 'rightarrow',

representing left and right pointing arrows. The difference between the two files is the way in which they are loaded. The file 'ISprites' is loaded into the WIMP's common sprite pool using the command *ICONSPRITES, while the two sprites from the file 'Sprites' are loaded into a user sprite area set up by our program. This is to ensure that our specific sprites don't interface with any having the same name, but belonging to other applications.

!RUNIMAGE

The final file to create in the application directory is !RunImage. This is the actual application, and is a sizeable Basic program. The first part of this is given in listing 4, which should be entered and saved in the application directory (noting the uneven line numbering). At this stage, you can now enter the Desktop and open the disc containing the Notepad. You should see the application complete with icon in the filer window, and double clicking on this should cause it to install itself on the iconbar. Double clicking on the iconbar icon will open the Notepad window, which at present will be blank except for an arrow in each top corner. Pressing the menu button over the icon will bring up a quit option. Selecting this will remove the Notepad.

Next month we will add the text handling routines to the program, and start to explain how it actually works.

Listing 1

```
| !Boot file for RISC User Note Pad
| by David Spencer
| Set up file type data
Set File$Type_FEE Note Pad
Set Alias$@RunType_FEE Run <Obey$Dir>.!Run
%%*0
| (%% to ensure substitution takes place
| when file is run, and not when command
| is obeyed.)
| Load wimp's sprites
IconSprites <Obey$Dir>.!Sprites
```

Listing 2

```
| !Run file for RISC User Note Pad
| by David Spencer
| First bit as for !Boot
Set File$Type_FEE Note Pad
Set Alias$@RunType_FEE Run <Obey$Dir>.!Run
%%*0
IconSprites <Obey$Dir>.!Sprites
| Find our path
Set NotePad$Dir <Obey$Dir>
| Check memory allocation
WimpSlot -min 32K -max 32K
| Start program
Run <NotePad$Dir>.!RunImage %*0
| (Just one %, so parameters are
| substiuted before !RunImage is run.
| 'Run' at start is necessary to
| prevent filing system name being
| treated as a temporary filing system
| for the duration of the command.)
```

Listing 3

```
  10 REM >SprGen
  20 REPEAT READ file$
  30 IF file$<>"*" THEN
  40 X%=OPENOUT file$
  50 READ len,check:sum=0
  60 REPEAT READ data$
  70 REPEAT
  80 byte=EVAL("&"+LEFT$(data$,2))
  90 sum+=byte:BPUT #X%,byte
 100 len-=1:data$=MID$(data$,3)
 110 UNTILdata$=""
 120 UNTIL1en=0
 130 CLOSE #X%
 140 OSCLI "SETTYPE "+file$+" Sprite"
 150 IF sum<>check THEN PRINT "Data error
in ";file$
 160 ENDIF
 170 UNTILfile$="*"
 180 END
 190 :
 200 DATA !Sprites,780,40798
 210 DATA 020000001000000010030000800100000
 220 DATA 216E6F7465706164000000004000000
 230 DATA 100000000000000070000002C000000
 240 DATA 2C0000000C0000001177117711771177
 250 DATA 1177117711771177110000011771177
 260 DATA 1177117711771177117711771000000
 270 DATA 7777777777777777777777777777777
 280 DATA 770000007700000000000000000000
 290 DATA 000000007700000077008B000B00B0B
 300 DATA 00B0BBBB00BBBB0B770000007700BB00
 310 DATA 0BB00BB00B00B000000B000077000000
 320 DATA 77000B0B0BB00BB00B00B00000BBBB00
 330 DATA 770000007700BB00BB00BB00B00B000
 340 DATA 000B0000770000007700BB00B00B00B
 350 DATA 0000B00000BBBB0B770000007700000
 360 DATA 000000000000000000000077000000
 370 DATA 770000008808000008008088000000000
 380 DATA 770000007700000008800808080008000
 390 DATA 080000707700000077000008808008 8
 400 DATA 880880000800002777000000770000000
 410 DATA 080000800088088000070227700000
 420 DATA 77777777777777777777777772722
```

```
430 DATA 77000000772222222222222222222222
440 DATA 22222222770000007777777777777777
450 DATA 77777777777777777700000080010000
460 DATA 66696C655F6665650000000004000000
470 DATA 10000000000000000700000002C000000
480 DATA 2C0000000C0000001177117711771177
490 DATA 11771177117711771100000011771177
500 DATA 11771177117711771177117711000000
510 DATA 77777777777777777777777777777777
520 DATA 77000000770000000000000000000000
530 DATA 00000000770000007700DDD0DD0DDDDD
540 DATA 0D0D0DDDDD0D000077000000770000000
550 DATA 00000000000000000000000077000000
560 DATA 77000DDDDDD00DD00DDD0DD0DDDDD0D00
570 DATA 77000000770000000000000000000000
580 DATA 00000000770000007700DDDD0DD0DD00
590 DATA DDDD0DDD0D0D000077000000770000000
600 DATA 00000000000000000000000077000000
610 DATA 7700DD0D00DDDDDDDD0DD0DD0DDDDD00
620 DATA 77000000770000000000000000000000
630 DATA 00000077070000007700DDDDDD00D0DD
640 DATA 0D00DDDD0D000D27770000007700000
650 DATA 00000000000000000000702277000000
660 DATA 77777777777777777777777772722
670 DATA 77000000772222222222222222222222
680 DATA 22222222770000007777777777777777
690 DATA 77777777777777777000000
700 DATA Sprites,228,9963
710 DATA 020000001000000E80000006C000000
720 DATA 6C6566746172726F7700000001000000
730 DATA 07000000000000001F0000002C000000
740 DATA 2C0000000C0000000707700000000000
750 DATA 00777700000000007077777777777777
760 DATA 77777777777777777077777777777777
770 DATA 00777700000000000707700000000000
780 DATA 00000000000000006C00000072696768
790 DATA 746172726F77000001000000070000000
800 DATA 000000001F0000002C0000002C000000
810 DATA 0C0000000000000007707000000000000
820 DATA 00777700777777777777777077777777
830 DATA 77777777777777777777700000000000
840 DATA 00777700000000000770700000000000
850 DATA 00000000
860 DATA *
```

Listing 4

```
  10 REM >$.!NotePad.!RunImage
  20 REM Program    Note Pad Utility
  30 REM Version    A 1.0
  40 REM Author     David Spencer
  50 REM RISC User April 1989
  60 REM Program    Subject to Copyright
  70 :
 140 maxpage=7:DIM block 200, mt 100, s
t 256, data (600*(maxpage+1)), menu 150,
 fi 100, pno 20
 150 $mt="<Untitled>":$st="Save as:"
 170 $block="TASK":SYS "Wimp_Initialise
",200,!block,"Note Pad" TO ,task
 180 ON ERROR PROCerror
 210 S%=OPEN"IN "<NotePad$Dir>.Sprites":
T%=EXT#S%+16:CLOSE #S%
 220 DIM sp T%:!sp=T%:sp!4=0
 230 sp!8=16:sp!12=16
 240 SYS "OS_SpriteOp",&10A,sp,"<NotePa
d$Dir>.Sprites"
 250 X=RND(624)-1:Y=RND(480)-1
 260 main=FNcreate(X,Y,656,544,&FF00000
2,0,656,544,3,mt)
 280 sicon=FNspriteicon(-1,0,0,68,68,&3
102,1,"!notepad")
 290 left=FNspriteicon(main,8,-36,40,-4
,&3102,sp,"leftarrow")
 300 right=FNspriteicon(main,616,-36,64
8,-4,&3102,sp,"rightarrow")
 350 quit=FALSE:tx=0:ty=0
 360 open=FALSE
 370 REPEAT
 380 SYS "Wimp_Poll",1,block TO reason
 390 CASE reason OF
 400 WHEN 1:PROCredraw(!block)
 410 WHEN 2:SYS "Wimp_OpenWindow",,bloc
k
 420 WHEN 3:PROCclose(!block)
 430 WHEN 6:PROCbuttons(block)
 460 WHEN 9:PROCmenuselect(block)
 470 WHEN 17,18:PROCreceive(block)
 480 ENDCASE
 490 UNTIL quit
 500 SYS "Wimp_CloseDown"
 510 END
 520 :
 530 DEF PROCreceive(b)
 540 CASE b!16 OF
 550 WHEN 0:quit=TRUE
 560 WHEN 2:PROCdatasave(b)
 570 WHEN 3,5:IF b!12=0 THEN PROCdatalo
ad(b)
 580 ENDCASE
 590 ENDPROC
 600 :
 610 DEF FNcreate(vx,vy,w,h,flags,bcol,
x,y,waf,title)
 620 $block=STRING$(88,CHR$0)
 630 !block=vx:block!4=vy
 640 block!8=vx+w:block!12=vy+h
 650 block!24=-1:block!28=flags
 660 block?32=7:block?33=2
 670 block?34=7:block?35=bcol
 680 block?36=3:block?37=1
```

Dabhand User News

Archimedes Basic Compiler Version 2 • Archimedes Operating System Guide
Special Offers on Archimedes PC Emulator and ANSI C

ABC Version 2

Don't live on promises, buy the only true BASIC V Compiler currently available, and now in its second release! But don't take our word here's what the reviewers said:

"...Excellent Dabs Press product. Buy it!" RISC User Dec.88

"ABC is a vital part of any programmer's toolbox, it puts compilers on other systems to shame. Unquestionably one of the most impressive pieces of software I have yet seen running on the Archimedes." A&B Dec 1988

"...I can tell you now, I am very impressed. This is a superb package." Archive Dec 1988

The above quotes were referring to Version 1 of ABC – ABC Version 2 is even better! Version 2 allows use of double and extended precision floating point, multiple exits from procedures and functions, RETURN parameter passing, new compiler directives and very much more.

As all reviewers have found, ABC makes writing machine code programs and relocatable modules easy. Programs can be written using the full range of BASIC V's error messages and report's, and once fully working, run through ABC which transforms them into machine code, ready for immediate action.

Demo Disc: We have produced a demo disc of ABC which still supports over 100 commands. It costs just £2 and this is refundable on any subsequent purchase. A full specification sheet is also available on request.

Free upgrades! **£99.95**

Archimedes OS Book

Hot off the presses, this latest Dabhand Guide is essential reading for every user of the Operating System including Arthur *and* RISC OS.

Its 320 pages are packed with vital information and covers topics such as VIDC, MEMC, IOC, Sound, SWIs, Vectors, Filing System and very much more. Price is just **£14.95** or **£21.95** with programs disc and manual.

Archimedes Assembly Language

386 pages devoted to programming the Archimedes in machine code. At **£14.95** it is the *only* book which deals specifically with assembler on the Archimedes. Book and programs disc ordered together – **£21.95.**

Archimedes Games!

Arcendium: Board Game Fun

Backgammon, Draughts, Reversi and Ouadline – four games for the price of one! Sure to get the whole family using the Archimedes. **Just £14.95.**

Alerion: Archimedes Arcade Action!

The highly acclaimed all-action shoot-em-up for the Archimedes. 256 colour mode graphics, with digitised speech. Impossible to finish! Superb fun!

"...the gameplay makes this game a winner...a first rate game..." A&B Computing. **Price £14.95.**

C: A Dabhand Guide

PCW said: *"I only wish this book had been available when I was learning C."* If you want to learn ANSI C then this 512 page volume is the way to do it. At **£14.95** it represents quite incredible value. Book and programs disc **£21.95.**

Archimedes PC Emulator Shareware

Five discs full of PC software tested with the emulator to ensure compatibility. The collection includes software you would normally expect to pay a fortune for and includes a wordprocessor, spreadsheet, games, flowchart designer, printer utilities, and more. **£34.95**

SPECIAL OFFERS! PC Emulator and FREE Shareware

Purchase the Archimedes PC Emulator from us and we'll give you our Shareware pack–detailed above and worth £34.95 – absolutely free of charge. Price just **£113.85** inc.

ANSI C and FREE Dabhand Guide to C

The best combination! Purchase Acornsofts ANSI C and we'll throw in our top-selling Dabhand Guide to C – 512 pages and worth £14.95. Price just **£113.85** inc VAT & p&p.

On the Lookout

Dabs Press are always on the lookout for authors who feel that they have what it takes to write either software or Dabhand Guides. If you have a useful idea then we can make it reality. If you don't have the idea but do have what it takes then we still want to hear from you. Our experience is second to none – if you want to be tops get in touch.

FREE on Request Bumper Catalogue!

DABS PRESS

5 Victoria Lane (RUM), Whitefield, Manchester M25 6AL
Tel: 061-766 8423 BT Gold: 72:MAG11596 Prestel: 942876210

Prices include VAT and P&P (UK/BFPO/CI). **ACCESS/VISA** accepted by post/phone/Mailbox/ in person. Cheques and POs to address above. Dabs Press products available from all good dealers. Add £2.50 (£12 air) if outside UK. Official orders welcome.

```
  690 block?38=12
  700 block!44=-y:block!48=x
  710 block!56=&13D:block!60=waf<<12
  720 block!72=title:block!76=-1
  730 block!80=LEN$title
  740 SYS "Wimp_CreateWindow",,block TO
handle
  750 =handle
  760 :
  770 DEF PROCredraw(handle)
  780 IF handle=main THEN
  790 !block=handle
  800 SYS "Wimp_RedrawWindow",,block TO
more
  810 WHILE more
  830 SYS "Wimp_GetRectangle",,block TO
more
  840 ENDWHILE
  850 ENDIF
  860 ENDPROC
  870 :
  880 DEF PROCbuttons(b)
  890 IF b!8 AND 2 THEN
  900 PROCmenu(b)
  910 ELSE
  920 CASE b!12 OF
  930 WHEN -2:REM Iconbar
  940 IF (b!16=sicon) AND NOTopen THEN
  950 !block=main
  960 SYS "Wimp_GetWindowState",,block
  970 SYS "Wimp_OpenWindow",,block
  980 open=TRUE
  990 ENDIF
 1260 ENDCASE
 1270 ENDIF
 1280 ENDPROC
 1290 :
 1300 DEF PROCclose(handle)
 1310 IF handle=main THEN open=FALSE
 1320 !block=handle
 1330 SYS "Wimp_CloseWindow",,block
 1340 ENDPROC
 1460 :
 1470 DEF FNspriteicon(window,x0,y0,x1,y
1,flags,sptr,sname$)
 1480 DIM block!24 LENsname$+1:block!28=
sptr
 1490 block!32=LENsname$+1:$block!24=sna
me$
 1500 =FNiconblk(window,x0,y0,x1,y1,flag
s)
 1510 :
 1570 DEF FNiconblk(window,x0,y0,x1,y1,f
lags)
 1580 !block=window
 1590 block!4=x0:block!8=y0
 1600 block!12=x1:block!16=y1
 1610 block!20=flags
 1620 SYS "Wimp_CreateIcon",,block TO ic
on
 1630 =icon
 1640 :
 1650 DEF PROCmenu(b)
 1660 CASE b!12 OF
 1670 WHEN -2:PROCdrawmenu(1,"Note Pad,Q
uit")
 1680 SYS "Wimp_CreateMenu",,menu+4,!b-6
4,136
 1790 ENDCASE
 1800 ENDPROC
 1810 :
 1820 DEF PROCdrawmenu(handle,menu$)
 1830 !menu=handle:menu?16=7
 1840 menu?17=2:menu?18=7:menu?19=0
 1850 menu!20=156:menu!24=40
 1860 menu!28=0:ptr=menu+32
 1870 $(menu+4)=LEFT$(menu$,INSTR(menu$,
",")-1)
 1880 menu$=menu$+","
 1890 WHILE menu$<>""
 1900 menu$=MID$(menu$,INSTR(menu$,",")+
1)
 1910 IF menu$<>"" THEN
 1920 !ptr=0:ptr!4=-1
 1930 ptr!8=&7000021
 1940 $(ptr+12)=LEFT$(menu$,INSTR(menu$,
",")-1)
 1950 ptr+=24
 1960 ENDIF
 1970 ENDWHILE
 1980 ptr!-24=ptr!-24 OR &80
 1990 ENDPROC
 2000 :
 2010 DEF PROCmenuselect(b)
 2020 CASE !menu OF
 2030 WHEN 1
 2040 CASE !b OF
 2050 WHEN 0:quit=TRUE
 2060 ENDCASE
 2140 ENDCASE
 2150 ENDPROC
 2160 :
 2170 DEF PROCerror
 2180 SYS "Wimp_DragBox",,-1
 2190 !block=ERR
 2200 $(block+4)=REPORT$+" (Internal err
or code "+STR$ERL+")"+CHR$0
 2210 SYS "Wimp_ReportError",block,1,"No
te Pad"
 2220 GOTO 370
```

RU

ARCHIMEDES SOFTWARE

(NOW RISC OS compatible)

SYSTEM DELTAPLUS (v2) £79.95

This Database Management System is the ultimate in simplicity and flexibility. Up ta 2 billian recards. Multiple files apen at any time. Multi-tasking allawing ane file ta be edited whilst searching anather. Search by simple selectian fram within windaws. Camprehensive maths, searches and sarts. Mailing labels, reparts, card capier.

ATELIER £99.95

(Ta be released saan).
Prafessianal quality art package. Functians include anti-alias squashing, ratatian af areas inta any quadrangle, and sprite and brush patterning. Further features allaw TV-style techniques ta even wrap items araund 3D surfaces. (Multi-tasking if using RISC OS.)

MAILSHOT £39.95

Versatile utility ta praduce mail-merged letters and labels acrass a sheet. (This is an add-an utility ta the System Delta Plus databave.)

REPORTER £39.95

Fully relatianal user defined reparting package. Allaws sub- tatals and sub-sub-tatals. Cantains mare facilities than dBASE III repart writer! (This is an add-an utility ta the System DeltaPlus database.)

SIGMASHEET £69.95

Massive fast spreadsheet. Definable calaurs, spacing, widths, farmats, etc. Over 30 maths/stats functians. Vertical/harizantal ar autamatic calculatian mades.

GAMMAPLOT £69.95

Extensive presentatian graphics/charts with art package. Multiple charts/graphs an screen anywhere. Pens, shapes, patterns, fants, black aptians, zaam, slide shaw facility. Imparts pictures/lagas. Merges with Wardpracessars.

Prices inclusive of VAT.

MINERVA

69 SIDWELL STREET, EXETER. EX4 6PH
Tel: (0392) 437756 Fax: (0392) 421762

Access & Barclaycard accepted

ACORN'S RAY TRACER

Roger Wilson, designer of the ARM chip, presents the program used to create the famous molecule demo.

This program requires RISC OS.

Almost everybody must have seen the rotating molecule demonstration produced by Acorn. If not, it was included in two parts on the RISC User discs for Volume 1 Issues 5 and 6. While very impressive, this demo only plotted a set of frames previously created using another program. This used a technique called *Ray Tracing* to work out the appearance of the image at any instant, and it is this basic program that is given here. In fact, the program is fairly general, and the version here produces a series of frames which can subsequently be animated to produce a rotating letter 'A' made up of coloured spheres. One frame from the sequence is, actually, one of the sample sprites supplied with RISC OS.

The first frame of the sequence

THE PROGRAM

The program from listing 1 should be typed in and saved before running it. The sprites generated as the program runs are particularly memory hungry, and you might need to perform some reconfiguring to increase application workspace before the program will run. The program will not run on a 305.

When the program is run, you will be prompted for a start frame, which should be a number between 0 and 129 (usually 0). The program will then proceed to calculate each frame, drawing it as it goes. Before each frame is drawn, a rough view of its shape is plotted in grey. This is the image projected forward onto the two dimensional screen. At the top of the screen you will see the frame number, the time taken on this frame, and the estimated time to complete the frame. The time to completion will initially rise, and then steady off. This is because it is calculated from the amount so far done, and the time taken. However, each pixel in the frame does not take exactly the same time to be traced. Once frame 129 has been calculated, the entire sprite file (about 550K long) is saved with the name 'RaySprites'.

The entire 130 frames take several hours to draw. If you press Escape, the program saves the current position in a file called 'StopData', and also the current sprites in the file 'RaySprites'. If when the program is re-run, you specify a negative start frame, the stop position is reloaded and the process continues.

Once the program has run, you will have a 130 frame sequence on disc. This is in the form of a single sprite file containing sprites named 000, 001, 002 ... 129. To animate the sequence, the program from listing 2 can be used. Next month we will feature a program that displays the animation sequence in a multi-tasking window, automatically rescaling it to the size of the window.

HOW IT WORKS

The explanation which follows concentrates on the principles involved, rather than the detailed operation of this long and complex program. If you want to find out more, an excellent book on this, and all aspects of computer graphics, is: *Procedural Elements for Computer Graphics* by David F. Rogers published by McGraw Hill in the International student series.

Ray tracing is very much a brute-force method of rendering an image (rendering is the process of making a conceptual object appear solid). Ray tracing works by considering the plight of every ray emitted from a light source, and this is done by taking each ray, and

reflecting it off the object by following the laws of physics and optics. When a ray hits the screen, the appearance of the particular pixel it hits can be determined. In practice, this proves to be very time-consuming, because the majority of rays from the light source will not pass through the screen. Therefore, in practice ray tracing is performed in reverse, by starting with each pixel on the visible screen and tracing it backwards. Each ray is reflected off surfaces of the object until it reaches the light source. This automatically caters for shadowing and reflections, but not refraction.

The way in which a ray is reflected off an object is determined by a number of factors, and the program uses a complex expression to model this. Basically, the ray is reflected about the normal to the surface at the point of intersection, with an intensity determined by the reflection model and the cosine of the solid angle between the reflected ray and the vector from the surface to the light source. Again, the recommended book explains this in great detail. To add more realism, the specular reflection model, as this function is called, allows for ambient light, that is, overall light that comes from no particular direction. However, ray tracing cannot in any way cope with the reflection of ambient light - an effect which occurs in real life.

MODIFYING THE PROGRAM

Despite its complexity, it is relatively easy to modify the program to handle any image made up of a series of spheres. The procedures which must be altered are PROCinit and PROCSetSpheres.

PROCinit sets up a number of parameters relating to the overall sequence. The three elements of the array *ls1* contain the x, y and z components of the position of the light source. The variable *maxframe* should be set to the total number of frames minus one. The values of *width* and *height* determine the size of area traced (in actual numbers of pixels), while increasing the value of *margin* limits the viewing area, with an obvious speed increase. *VD* is the distance of the viewer from the object. *NoSpheres* is the number of spheres that make up the object, and finally, *res* is the resolution to which the object is traced. A value greater than one gives a faster trace, but with a loss of definition. You can experiment to see the exact effect of these values.

PROCSetSpheres is the routine that actually *draws* the object in 3-space. This routine is called each time a frame is about to be rendered, and is passed the current frame number. The routine must then create the object by setting up the contents of twelve arrays, described below. The change of object from frame to frame must also be catered for in this routine. If you look at the program, you will see that it starts by drawing the base image into the arrays, and then applying transformations to it according to the frame number.

Each sphere is represented by a set of corresponding elements in the twelve arrays, the array subscript being the sphere number. The arrays are:

x( )	The position of the centre of the
y( )	sphere in 3-space
z( )	
sz( )	The *square* of the radius of the sphere
lr( )	The red, green and blue
lg( )	components of the colour of the
lb( )	sphere. These are real values between 0 and 1.
ka( )	Ambient reflectance
kn( )	Normal illumination
ksp( )	Coefficients of specular
kr( )	reflection
ks( )	

These last five arrays are the parameters used for the illumination and specular reflection models. Their meanings are fully explained in the recommended book, but I suggest that unless you understand them fully, you stick to the values used in the program. This will give the spheres an appearance not dissimilar to that of vinyl silk paint. If you follow the SetSpheres procedure in the program, the above explanation should become clearer.

There are many other ways in which you might like to try and modify the program. For

example, you could improve the illumination model, or allow for more than one light source, or even allow for the effects of refraction which would make transparent objects possible. If you really want a challenge then you could make the program cater for non-spherical objects, but be warned, this is not a trivial task!

A frame from the middle 'spin' phase

This month's magazine disc contains the two programs given here, and also the version of the program used to produce the famous molecule.

Listing 1

```
   10 REM > Trace
   20 REM Program    Ray Tracer
   30 REM Version    A 1.00
   40 REM Author     Roger Wilson
   50 REM RISC User April 1989
   60 REM Program    Subject to Copyright
   70 :
   80 MODE 13:OFF:ORIGIN 512,320
   90 DIM ls1(2),ls2(2),l(2),ol(2),p(2),
np(2),q(2),n(2),s(2),colour(2)
  100 PROCinit
  110 DIM SA% SASIZE%:SA%!0=SASIZE%
  120 SA%!4=0:SA%!8=16:SA%!12=16
  130 resX%=4*res: resY%=4*res
  140 DIM sz(NoSpheres-1)
  150 DIM lr(NoSpheres-1)
  160 DIM lg(NoSpheres-1)
  170 DIM lb(NoSpheres-1)
  180 DIM ka(NoSpheres-1)
  190 DIM kn(NoSpheres-1)
  200 DIM ks(NoSpheres-1)
  210 DIM ksp(NoSpheres-1)
  220 DIM kr(NoSpheres-1)
  230 DIM x(NoSpheres-1)
  240 DIM y(NoSpheres-1)
  250 DIM z(NoSpheres-1)
  260 :
  270 DIM qX(NoSpheres-1),qY(NoSpheres-1
),qZ(NoSpheres-1)
  280 DIM  t(NoSpheres-1), b(NoSpheres-1
), c(NoSpheres-1)
  290 SO%=&2E:SO_New=&109:SO_Load=&10A
  300 SO_Save=&10C:SO_Get=&110
  310 INPUT"Start Frame ";SF
  320 IFSF<0 THEN
  330 A%=OPENIN"StopData"
  340 INPUT#A%,SF
  350 CLOSE#A%
  360 PROCTMDLOAD("RaySprites")
  370 ENDIF
  380 PROCASS
  390 ToSprite%=FALSE
  400 ON ERROR PROCCAPTURE
  410 linetime%=0
  420 FOR frame=SF TO maxframe
  430 PROCHDR
  440 Start=TIME
  450 PROCSetSpheres
  460 PROCProjectScene
  470 PROCDrawScene:PRINTTIME-Start
  480 SpriteName$=RIGHT$("000"+STR$frame
,3)
  490 SYS "OS_SpriteOp",SO_Get,SA%,Sprit
eName$,0,0,0,252,252
  500 NEXT
  510 PROCTMDSAVE("RaySprites")
  520 END
  530 :
  540 DEF PROCCAPTURE
  550 ON ERROR OFF
  560 IFERR<>17 REPORT:PRINT" at line ";
ERL:END
  570 MODE0
  580 A%=OPENOUT"StopData"
  590 PRINT#A%,frame
  600 CLOSE#A%
  610 PROCTMDSAVE("RaySprites")
  620 END
  630 :
  640 DEF PROCDrawScene
  650 FOR Y%=BM TO TM STEP resY%
```

```
  660 PRINT TAB(cx%,cy%);Y%
  670 IF 1inetime% THEN PRINT"Estimated
time to completion=";((TM-Y%)DIVresY%+1)
*1inetime%DIV100" sec.    "
  680 PRINT"Doing this frame for ";(TIME
-Start)/100" sec.    "
  690 T1%=TIME
  700 FOR X%=LM TO RM STEP resX%
  710 IF POINT(X%,Y%) THEN
  720 PROCget1p(X%,Y%)
  730 PROCcheck:IFH%THEN
  740 colour(0)=2/16:colour(1)=8/16
  750 colour(2)=8/16
  760 ELSE
  770 k=0:colour()=0:ik=1
  780 REPEAT
  790 PROCreflect:k+=1
  800 PROCcheck
  810 UNTIL H% OR k>40
  820 ENDIF
  830 PROCplot
  840 ELSE
  850 PROCplot2
  860 ENDIF
  870 NEXT
  880 1inetime%=(1inetime%+(TIME-T1%))/2
  890 NEXT
  900 ENDPROC
  910 :
  920 DEF PROCget1p(X,Y)
  930 REM move pixel on view screen to i
mage space
  940 l(0)=X-XC:l(1)=Y-YC:l(2)=VD
  950 p(0)=l(0):p(1)=l(1):p(2)=0
  960 l()=l()/MODl()
  970 ENDPROC
  980 :
  990 DEF PROCSetSpheres
 1000 REM initialise image space
 1010 RESTORE
 1020 IFframe<40 an=frame/40*PI:tilt1=0
ELSE an=PI
 1030 IFframe>39 IF frame<90 tilt1=(fram
e-40)/50*PI
 1040 IFframe>89 tilt1=PI:tilt2=(frame-9
0)/40*PI ELSE tilt2=0
 1050 factor=34 : REM approx factor to c
alculate atomic radii
 1060 N=0:C=1:RESTORE
 1070 FORY=6TO0STEP-1
 1080 X=0:READA$:FORJ%=1TOLENA$
 1090 IFMID$(A$,J%,1)="1" THEN
 1100 x(N)=X*40-120:y(N)=Y*40-120:z(N)=0
 1110 CASE C OF
 1120 WHEN 1:lr(N)=1:lg(N)=.25:lb(N)=.25
 1130 WHEN 2:lr(N)=.25:lg(N)=1:lb(N)=.25
 1140 WHEN 3:lr(N)=1:lg(N)=1:lb(N)=.25
 1150 WHEN 4:lr(N)=.25:lg(N)=.25:lb(N)=1
 1160 WHEN 5:lr(N)=1:lg(N)=.25:lb(N)=1
 1170 WHEN 6:lr(N)=.25:lg(N)=1:lb(N)=1
 1180 WHEN 7:lr(N)=1:lg(N)=1:lb(N)=1
 1190 ENDCASE
 1200 ka(N)=.2:kn(N)=.6:ks(N)=.55:ksp(N)
=30:kr(N)=1/3
 1210 sz(N)=factor^2
 1220 N+=1:C+=1:IFC>7 C=1
 1230 ENDIF
 1240 X+=1:NEXT
 1250 NEXT
 1260 cosan=COSan:sinan=SINan
 1270 cost2=COStilt1:sint2=SINtilt1
 1280 IFframe>39IFframe<90 cost2a=COS(2*
PI-tilt1):sint2a=SIN(2*PI-tilt1) ELSE co
st2a=cost2:sint2a=sint2
 1290 cost1=COStilt2:sint1=SINtilt2
 1300 FORN=0TONoSpheres-1
 1310 t1=x(N)*cosan-z(N)*sinan:REM spin
 1320 t2=x(N)*sinan+z(N)*cosan
 1330 x(N)=t1:z(N)=t2
 1340 t1=y(N)*cost1-z(N)*sint1:REM tilt
 1350 t2=y(N)*sint1+z(N)*cost1
 1360 y(N)=t1:z(N)=t2
 1370 IFN AND1 cos2=cost2:sin2=sint2 ELS
E cos2=cost2a:sin2=sint2a
 1380 t1=y(N)*cos2-x(N)*sin2:REM rotate
 1390 t2=y(N)*sin2+x(N)*cos2
 1400 y(N)=t1:x(N)=t2
 1410 NEXT
 1420 z()+=1000
 1430 ENDPROC
 1440 DATA 0000001
 1450 DATA 0000011
 1460 DATA 0000101
 1470 DATA 0001111
 1480 DATA 0010001
 1490 DATA 0100001
 1500 DATA 1000001
 1510 :
 1520 DEF PROCProjectScene
 1530 REM Show gray shadow of scene
 1540 GCOL 0,32+8+2 TINT 0:k=VD
 1550 FORN=0 TO NoSpheres-1
 1560 z=z(N)+VD:CIRCLE FILL k*x(N)/z+XC,
k*y(N)/z+YC,k*SQR(sz(N))/z+4
```

```
 1570 NEXT
 1580 ENDPROC
 1590 :
 1600 DEF PROCplot
 1610 REM plot final coloured pixel
 1620 IFABS(colour(0)-colour(1))<1/32AND
ABS(colour(1)-colour(2))<1/32THEN
 1630 r%=FNB(colour(0)*16):g%=r%:b%=r%:t
%=r%
 1640 ELSE
 1650 r%=FNB(colour(0)*16)
 1660 g%=FNB(colour(1)*16)
 1670 b%=FNB(colour(2)*16)
 1680 IFABS(colour(0)-colour(1))<1/32 r%
=g%
 1690 IFABS(colour(0)-colour(2))<1/32 r%
=b%
 1700 IFABS(colour(1)-colour(2))<1/32 b%
=g%
 1710 t%=g%:IFt%+1<r% t%=r%
 1720 IFt%+1<b% t%=b%
 1730 ENDIF
 1740 GCOL((r%AND12)>>2)+(g%AND12)+((b%A
ND12)<<2) TINT t%<<6
 1750 IFresX%=1POINTX%,Y% ELSERECTANGLEF
ILLX%,Y%,resX%-4,resY%-4
 1760 ENDPROC
 1770 :
 1780 DEF FNB(X) LOCALX%:X%=INTX:IFRND(1
)<X-X% X%+=1
 1790 IFX%>15X%=15
 1800 =X%
 1810 :
 1820 DEF PROCplot2
 1830 REM plot background colour pixel
 1840 GCOL40 TINT 0
 1850 IFresX%=1POINTX%,Y% ELSERECTANGLEF
ILLX%,Y%,resX%-4,resY%-4
 1860 ENDPROC
 1870 :
 1880 DEF PROCcheck
 1890 REM compute intersection of normal
ised vector l() from position p()
 1900 REM Returns H%=TRUE if no hit, els
e H%=FALSE, L=distance, M%=sphere number
 1910 H%=TRUE:qX()=p(0)-x():qY()=p(1)-y(
):qZ()=p(2)-z()
 1920 b()=l(0)*qX():t()=l(1)*qY():b()=b(
)+t():t()=l(2)*qZ():b()=b()+t()
 1930 c()=qX()*qX():t()=qY()*qY():c()=c(
)+t():t()=qZ()*qZ():c()=c()+t()
 1940 CALL CODE%,c(),sz(),b(),H%,L,M%
 1950 ENDPROC
 1960 :
 1970 REM Code performs:
 1980 FORN%=0TONoSpheres-1:IFt(N%)<0.0NE
XT:ENDPROC
 1990 t=SQRt(N%):K=-b(N%)-t:t-=b(N%):IFK
<t t=K
 2000 IFt<1E-3NEXT:ENDPROC
 2010 IFH%L=t:M%=N%:H%=FALSE ELSEIFt<L L
=t:M%=N%
 2020 NEXT:ENDPROC
 2030 :
 2040 DEF PROCreflect
 2050 REM compute new reflected ray. Als
o compute colour effects.
 2060 s(0)=x(M%):s(1)=y(M%):s(2)=z(M%)
 2070 q()=l()*L:np()=p()+q():n()=np()-s(
)
 2080 n()=n()/MODn()
 2090 q()=n()*l():q()=(2*SUMq())*n()
 2100 ol()=l():l()=lsl():d=L:M1%=M%:s()=
1E-4*n():p()=np()+n():PROCcheck
 2110 IFH%=TRUE THEN
 2120 s()=n()*l():s()=(2*SUMs())*n():l()
=l()-s():l()=l()/MODl()
 2130 s()=n()*lsl():Bl=kn(M1%)*SUMs():IF
Bl<0 Bl=0
 2140 s()=l()*ol():B2=ks(M1%)*SUMs()^ksp
(M1%):IFB2<0 B2=0
 2150 s(0)=lr(M1%):s(1)=lg(M1%):s(2)=lb(
M1%)
 2160 s()=s()*(ka(M1%)+Bl):s()=s()+B2
 2170 ELSE
 2180 s(0)=lr(M1%):s(1)=lg(M1%):s(2)=lb(
M1%):s()=s()*ka(M1%)
 2190 ENDIF
 2200 s()=s()*ik
 2210 colour()=colour()+s():ik=ik*kr(M1%
)
 2220 l()=ol()-q():l()=l()/MODl():H%=FAL
SE:p()=np()
 2230 ENDPROC
 2240 :
 2250 DEF PROCASS
 2260 DIM CODE% 2500
 2270 P%=CODE%
 2280 [ OPT 0
 2290 STR R14,[R0]:MOV PC,R14
 2300 ]:A%=CODE%+20:CALLCODE%
 2310 F1STA=CODE%!20+22*4:F1LDA=F1STA+4
 2320 F1ADD=F1LDA+4:F1XSUB=F1ADD+4
```

Continued on page 47

ANNOUNCING A MAJOR NEW SOFTWARE RELEASE

Premier

Circle Softwere ere proud to ennounce e mejor new text processor for the Archimedes feeturing e mejor edvence in capebility. Premier is the first progrem of it's kind to include whet we heve called -

DYNAMIC TEXT

This new concept ellows the inclusion of items within eny document, whose textuel content is not fixed, but mey chenge efter the item hes been inserted. Such DYNAMIC TEXT items mey show todey's dete or time, the current velue of VAT, or e neme end eddress. The ectuel text displeyed mey be defined et the keyboerd, or in e seperete text file. In fect different text files mey define these DYNAMIC TEXT items differently, to produce stetements end invoices, finenciel reports, or personelised mell ehots.

This is no pipe dreem. Premier's powerful command lenguege includes full meths functions, repeet loops end 'if - then' constructs, ell of which mey be included in the definition of e DYNAMIC TEXT item, ellowing full spreed-sheet type feclities enywhere on the pege, not just in neet rows end columns. But Premier does not stop there.

The ebility to reed end execute commends from text files ellows eny text processing to be carried out just by typing a 'ster' commend. Further, the ability to define 'mecro' commends, plus the ability to define the ection of ANY key, ellows complete 'epplications' to be built, including customised spreedsheets, end simple dete beses. A commend file defining e full feetured word processor, end e metching key-strip, is included with Premier.

This word processor elone is e justifiable reeson for the purchese of Premier. It offers four wey scrolling windows, is eble to cut end peste text between eny number of files, feetures forwerd end beckwerd seerches, seerch pettern metching, headers end footers employing DYNAMIC TEXT, end other feetures too numerous to list here. The commend file defining ell this mey be edited, to teilor the feetures to your own requirements.

Premier will be enhenced further to include more feetures end epplications. As e RISC USER reeder you can receive ell updetes free during 1989 by ordering NOW et e speciel price of just 50 pounds, or see DYNAMIC TEXT in ection end try our word processor on e demo disc evieleble for just 5 pounds, refundeble on e full order.

Send NOW for full deteils of Premier,
e refundable demo disc for 5 pounds,
or e full version of Premier, complete
with menuel, key strips, end e no quibble
guerentee, for just 50 pounds.
You heve nothing to loose!

Circle Software

33, Restrop View,
Purton, Swindon,
Wilts. SN5 9DG

THE RISC OS HOURGLASS

David Spencer looks at a useful feature provided by RISC OS.

The RISC OS *Hourglass* module is a short module that can be used to pop-up an egg timer on the screen as a mouse pointer. Its main use is to inform the user that the machine is tied up at the present time, and will not respond to keyboard input. The hourglass often appears when a directory is first opened using the Desktop.

Controlling the hourglass is simplicity itself. All controls are in the form of SWI commands, which are issued from Basic using the SYS statement. The two most important commands are:

```
SYS "Hourglass_On"   and
SYS "Hourglass_Off"
```

As their names suggest, these are used to turn the hourglass on and off. The first command should be placed before any piece of code that takes a relatively long time to execute, and the second command afterwards, for example:

```
10 SYS "Hourglass_On"
20 PRINT "Start"
30 TIME=0
40 REPEAT UNTIL TIME>1000
50 PRINT "End"
60 SYS "Hourglass_Off"
```

This turns the hourglass on, prints 'Start', waits ten seconds, prints 'End' and finally turns the hourglass off. Incidentally, using the hourglass corrupts the definitions of pointers 3 and 4, although this will not normally be a problem.

If you try the above program, you might notice a small delay between the 'Start' being printed, and the hourglass appearing. This isn't at all accidental - instead, when you turn the hourglass on, it doesn't actually appear for 1/3 second, although the SYS command returns to the program immediately. This is so that the hourglass is only displayed if the operation takes a noticeable time (more than 1/3 second). Otherwise, the hourglass is turned off before it is displayed, and never actually appears. If the delay of 1/3 second is too long or too short, an alternative command can be used:

```
SYS "Hourglass_Start",delay
```

This is the same as 'Hourglass_On", but instead of using the fixed delay, the delay is specified as a parameter (in 100ths of a second). To illustrate this, change line 10 of the above program to:

```
10 SYS "Hourglass_Start",200
```

Hourglass on and off calls are nested. This means that if you turn the hourglass on more than once (without turning it off), you must turn it off the corresponding number of times before it is no-longer displayed. This ensures that in the situation where one routine that uses the hourglass calls another which also uses it, the hourglass is displayed until the first routine has finished, and isn't accidentally turned off by the second. The command:

```
SYS "Hourglass_Smash"
```

can be used to remove the hourglass display immediately, regardless of how many times it was turned on. This is very useful in error handling routines.

INDICATORS

As well as the basic hourglass, it is also possible to display a percentage figure below the 'egg timer', and two LED displays (blue bars) , one above and one below. For example, if the hourglass was displayed while a word processor reformatted a document, it might be helpful to display the percentage of the text that has been processed. The call to do this is:

```
SYS "Hourglass_Percentage",percent
```

If the value of percent is in the range 0 to 99, then the percentage is displayed. Any other value turns off the percentage display. To continually update the percentage, your program must issue this call each time the value changes.

The LEDs are controlled using:

```
SYS "Hourglass_LEDs",eor,and
```

The current state of the LEDs is represented by a two-bit number, bit 0 is set if the top LED is on, and bit 1 if the bottom LED is on. This call takes the current state, ANDs it with the parameter *and*, and Exclusive ORs the result with the parameter *eor* to get the new state. This allows you to achieve effects such as inverting the state of an LED, or changing one while leaving the other alone. The simplest way to use this command is to ignore the and parameter, and use the first parameter (eor) to specify the state of the LEDs. For example:

```
SYS "Hourglass_LEDs",2
```

will turn the bottom LED on, and the top one off.

Lingenuity

Specialist Software

Archimedes *PRODUCTS FOR THE 1990's*

NEW PRESENTER STORY

The ultimate in Business Presentation software. Turn your Archimedes into a tool to project combined images of Text, Logos, Graphs, Digitised & Video Pictures to illustrate your presentations more clearly. Many special screen features are included such as vertical and smooth scrolling and bouncing. It can be used with most makes of RGB projectors or you can use your Archimedes monitor for smaller Presentations. PRESENTER STORY is available for Business applications now. Education and Media/Video versions will be available soon. RISCOS compatible.

£199.00 + VAT

PRESENTER

The established presentation package for Archimedes users in schools and businesses alike. Colour 3-D Bar-, Line- and Pie- charts and graphs can be made quickly and easily in the Acorn WIMP environment. These charts and graphs can be printed on FX-compatible and Integrex colour printers and also on Plotmate plotter. Alternatively, they can be ported into other packages, such as 1st Word Plus, Artisan & Graphic Writer. RISCOS compatible.

£24.95 + VAT

NEW CONTROL PANEL

*Confused about the *Configure comands? With Control Panel there is no need to worry. Use Control Panel to set the right Archimedes' configuration for all your application software. You can then Save and Load the configuration settings of your choice for use at later dates. Control Panel also includes a utility to build configuration files into program boot files. Fully wimp-driven with constant real-time help screen.*

£14.95 + VAT

NEW COLOUR CONVERTER

Do you want to capture colour images onto your Archimedes screen? The Colour Converter podule running with the Watford Digitiser allows full colour images to be captured and saved as screens or sprites. The package also includes new Dithering Software which will give you greatly enhanced near-television quality pictures, using an effective palette of 256,000 colours! The Colour Converter ½ Width Podule and Software is available now.

£169.95 + VAT

NEW A I M (Archimedes Image Manager)

Public Domain Image Processing software from Delft University which will increase the contrast and sharpen blurred images that have been digitised. Many varied uses – in education, industry and the home. A multisync monitor is required in order to give the necessary resolution.

£9.00 + VAT

VISA and ACCESS accepted.

All Lingenuity products are available direct or from good Acorn dealers.

Educational discounts and site-licences are available.

For further details please contact us on 098 685 477 or write to Lingenuity, PO Box 10, Halesworth, Suffolk IP19 0DX.

ART NOUVEAU

Reviewed by Mike Williams

A good many art and painting packages have appeared for the Archimedes since it was launched, not least ProArtisan from Clares Micro Supplies, reviewed in Volume 2 Issue 3. Readers may well be forgiven for wondering if there is room for another (and there are more to come).

Art Nouveau is clearly a quality product; even a brief familiarisation with the package will readily convince anyone of the amount of thought and development which has gone into this package. For your £42.50 you get a disc and a manual of some 100 pages or so. The software works in mode 15 (256 colours), though there is an option to load mode 12 screens and convert them to mode 15.

Product	Art Nouveau
Supplier	Computer Assisted Learning Ltd Strathclyde Business Centre, New Stevenston, Strathclyde, Scotland ML1 4JB. Tel. (0698) 733775
Price	£42.50 Inc.

For this review, Art Nouveau was used on an Archimedes 310 with RISC OS installed, and no problems were encountered with this combination. However, the software cannot be run directly from the Desktop, though this may well be provided for in production versions. At present, Art Nouveau starts by saving the current configuration of your machine which is then reconfigured. There appears to be no way of avoiding this, even if the machine has already been put through this process, and I do not feel totally happy with this approach.

Once loaded, Art Nouveau presents a totally black screen, apart from a white border. To make any use of the software it is essential to refer to the manual, so it is fortunate that this is well written. It appears to be a fact of life with the Archimedes that every software house adopts a different strategy for the use of pop-up menus and mouse control, and Art Nouveau is no exception, though it is logical and probably follows Acorn's suggested standards more than most.

Pressing the *menu* (middle) button at any time displays a menu bar across the top of the screen (with eight menu options), and a status display at its foot. Release the menu button and the menu bar immediately disappears. Even after several hours spent using this software, I still found this disconcerting. Moving the mouse pointer to any option on the menu bar triggers a pull-down menu, and many of the menu options either link through to further sub-menus, or call up a new screen for some major options. The menus are very stylish in appearance, and the whole system adopts a consistent approach.

LINE DRAWING

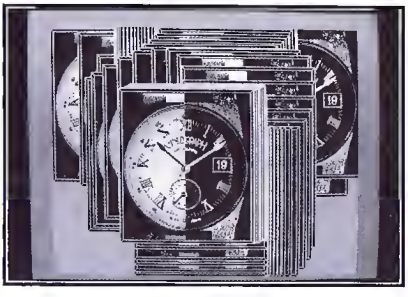
Two of the main menus provide for the drawing of straight or curved lines and shapes, with a choice of outline or filled shapes. Straight line drawing includes single and continuous rubber-band drawing, plus squares, rectangles, parallelograms and triangles. User defined polygons can be created, and the result positioned, scaled and rotated as required.

Art Nouveau

Curve drawing includes ellipses, circles, arcs, segments and sectors, and also a free-flowing curve joining two points. This is much like the Bezier curves used in some other software, but bends in one direction only ('C' not 'S' shaped). In practice it is simple to use, and a succession of such curves may be easily and smoothly joined together.

PENCILS AND BRUSHES

Freehand drawing uses the *pencil* option in the *goodies* menu. This can be used in one of two ways. In one case it leaves a trail of dots, and the extent to which this appears as a continuous line depends upon the speed with which drawing takes place. The other format always creates a continuous line irrespective of drawing speed. The mouse can also be *locked* so that lines can be accurately drawn vertically and horizontally.



Painting with a brush

Line drawing can also use a so-called colour cycle, in which the sequence of dots forming a line follows a pre-determined sequence of colours. Several such sequences are supplied on the Art Nouveau disc and may be loaded and selected as required. Alternatively the user may design his own colour cycles, and in turn save these to disc.

You can also select a *spray brush,* and control the shape and size of the 'droplets', as well as the size of the spray area. It is also worth saying here that Art Nouveau provides a *colour* merge option. This mixes the pixels within a small square box displayed on screen when the *action* (left-hand) mouse button is pressed. This needs to be used with care, as the results are not that subtle, and on the whole I would rate this less highly than ProArtisan's *wash* facility.

Quite different to either of the above is a *brush* facility. This 'loads' a brush with a rectangular image or pattern with which to paint on the screen. The size of a brush is not restricted other than by the size of the screen itself. A number of brushes are supplied as part of the Art Nouveau disc. Once a brush is selected, pressing the *action* button will place the brush image on the screen. The brush can also be used like a spray brush to leave a continuous trail of images.

Once a brush has been selected, its size can be halved or doubled, independently in the x and y directions, or stretched (and that means reduced as well) to any size required. However, the resulting image seems often coarser and less attractive than one generated at the required size initially.

Distorting brush images

Some beautiful effects can also be achieved by applying any of several distortions to a brush image, and there is an option to disort a brush onto any enclosed area. Any of the colours in a brush can also be made *transparent* so that the existing background shows through in these areas. This is very useful for creating a brush which appears not to have a rectangular shape but that of some intricate object.

Finally, the user can create and save his own design of brush. This can be done by

picking up any rectangular area of the current screen display, or (in order to preserve the current screen image), by switching to a separate *scratch* screen which can be used independently of the main drawing area. Any brush generated can then be saved to disc for future use.

Using the pixel editor

PIXELS, FONTS AND PATTERNS

Art Nouveau is an entirely pixel oriented package, and nowhere does this show more clearly than in the *Pixel* Editor. This is an excellent example of its kind, allowing the image to be enlarged 2, 4, 8, or 16 times for detailed editing of pixels.

Art Nouveau also allows text to be incorporated on a screen design, using the stndard Acorn font or an alternative supplied with Art Nouveau. There is also a font editor allowing the user to design and save his own fonts.

Various patterns are supplied both as default and with others on disc. Any colour or pattern can be selected for the flood-fill operation, and the screen can also be cleared to a pattern as well as a colour. A pattern editor can be used to generate new designs, which can then be saved to disc, and it is also possible to select the current brush as a pattern.

HOUSEKEEPING FUNCTIONS

Disc management seems excellent. The *disc* menu has the expected save and load functions, covering screens, brushes, and patterns, but will also format and initialise a new disc, and caters well for disc changes while using Art Nouveau. There is also an option to delete files, which can be very handy at times.

Two print options are provided, one for Epson FX compatibles with the facility to adjust the tonal range, and a colour dump which I tried with an Integrex Colourjet 132. The results of this seemed very washed out compared with the richness of the screen image, and colour prints from ProArtisan are better, though I have yet to see a reasonably priced colour printer which will achieve really good rendering of a 256 colour mode.

MANUAL

I have already referred to this. Nearly three-quarters of its contents are tutorial in style, and well written in a way which really does take you step by step through virtually all the facilities available. The remainder acts as a reference guide to the menu options. The quality of the illustrations was poor, but my copy was a draft version only. Having ploughed my way right through the tutorial section I was left feeling a little confused, and not clear about which menu provided which facility. If this is so, then it may derive more from the design of the menu system rather than from any inadequacies in the manual itself. Any criticsm here can also be taken as a compliment as it reflects the very richness of the Art Nouveau environment.

CONCLUSIONS

As indicated earlier, my copy of Art Nouveau was not a final version. However, it is certainly impressive, and deserves to sell well. It will take anyone some time to get used to the menu system, but this is true of many packages. Once past that hurdle, Art Nouveau will prove a powerful and versatile addition to the range of art packages for the Archimedes, and is a good first product by a new company, and at a highly competitive and attractive price. RU

BASIC PROGRAM LISTING UTILITY

To illustrate the use of Basic's token table, David Spencer presents a utility to list a program from a file.

The article elsewhere in this issue on the Basic token table shows how a token can be converted back into the corresponding keyword. This principle is used here to implement a utility to list a Basic program from a file, without having to first load it into memory. Two versions of the utility are given. The first is written in Basic, and has the advantage of being easy to follow if you wish to investigate how it works. The other version is in the form of a machine code utility which can be saved in the Library directory of your disc and used as required.

USING THE UTILITY

Using the Basic version of the utility is simplicity itself. Type in the program and save it, then run it and enter the file name of the program to be listed. The required file will then be listed just as if it had been loaded and LIST used.

To use the machine code version, type it in, save it and run it. This will assemble the utility and save it using the filename 'BLIST'. The command is invoked with:

```
*BLIST <file> [P]
```

The parameter *file* is the filename of the Basic program to list. By placing a 'P' after the filename, for example:

```
*BLIST Prog P
```

the command will pause until a key is pressed. This allows discs to be swapped if the program to be listed is on a different disc to the actual command. Note however, that in this case the filename must specify the drive number, otherwise ADFS will not remount the disc.

```
  10 REM             >FLister
  20 REM Program     Basic Program List
  30 REM Version     A 1.0
  40 REM Author      David Spencer
  50 REM RISC User April 1989
  60 REM Program Subject to Copyright
  70 :
  80 DIM code 100
  90 FOR pass=0 TO 2 STEP 2
 100 P%=code:[OPT pass
 110 STMFD R13!,{R14}:MOV R2,R14
 120 ADR R12,temp:STR R1,[R12]
 130 ADR R14,back:ADD PC,R2,#&4C
 140 .back STR R1,temp:LDMFD R13!,{PC}
 150 .temp EQUD 0:]NEXT
 160 @%=&505:ON ERROR GOTO 590
 170 INPUT "Enter filename "F$
 180 X%=OPENIN F$
 190 IF X%=0 ERROR &D6,"File not found"
 200 Y%=FNget
 210 IF Y%<>13 ERROR 0,"Not BASIC"
 220 REPEAT L%=FNget
 230 IF L%<>255 THEN
 240 ln=256*L%+FNget:Y%=FNget
 250 PRINT ln;" ";
 260 REPEAT Y%=FNget
 270 CASE TRUE OF
 280 WHEN Y%=13
 290 WHEN Y%=&8D:PROClref
 300 WHEN Y%>&7F:PROCtoken(Y%)
 310 OTHERWISE:VDU Y%
 320 ENDCASE
 330 UNTIL Y%=13
 340 PRINT
 350 ENDIF
 360 UNTIL L%=255
 370 CLOSE #X%
 380 END
 390 :
 400 DEF FNget
 410 IF EOF#X% THEN CLOSE#X%:END
 420 =BGET#X%
 430 :
 440 DEF PROCtoken(A%)
 450 IF A%>&C5 AND A%<&C9 THEN B%=FNget
 460 CALL code:addr=!temp
 470 WHILE ?addr<&7F
 480 VDU ?addr:addr+=1
 490 ENDWHILE
 500 ENDPROC
 510 :
 520 DEF PROClref
 530 no1=FNget<<2
 540 no2=FNget EOR (no1 AND &C0)
 550 no2=no2 OR ((FNget EOR ((no1<<2) A
ND &C0))<<8)
 560 PRINT;no2;
 570 ENDPROC
 580 :
 590 @%=&90A:IF X% THEN CLOSE#X%
 600 PRINT REPORT$ " at line ";ERL
```

```
**********************************************
  10 REM             > FileList
  20 REM Program     ARM program lister
  30 REM Version     A 1.0
  40 REM Author      David Spencer
  50 REM RISC User April 1989
  60 REM Program Subject to Copyright
  70 :
```

```
 80 DIM code 1000
 90 FOR pass=0 TO 2 STEP 2
100 P%=code
110 [OPT pass
120 LDR R0,[R14,#&4C]:TST R0,#&800000
130 BICEQ R0,R0,#&FF000000
140 ORRNE R0,R0,#&FF000000
150 ADD R1,R14,#&54
160 ADD R0,R1,R0,LSL #2
170 ADR R1,temp:STR R0,[R1]
180 MOV PC,R14
190 .temp EQUD 0:]
200 NEXT:CALL code
210 tokenaddr=!temp
220 :
230 FOR pass=4 TO 7 STEP 3
240 P%=0:O%=code
250 [OPT pass
260 SUB R13,R13,#256
270 STMFD R13!,{R14}
280 LDRB R0,[R1]:CMP R0,#ASC" "
290 LDMCCFD R13!,{R14}:ADRCC R0,err1
300 ORRCCS PC,R14,#1<<28
310 MOV R0,R1:MOV R2,#%101<<29
320 SWI "XOS_GSInit":MOV R3,R12
330 .readname SWI "XOS_GSRead"
340 BVS error2:MOVCS R1,#0
350 STRB R1,[R3],#1:BCC readname
360 SUB R0,R0,#2
370 .findq LDRB R1,[R0,#1]!
380 CMP R1,#32:BEQ findq:BCC open
390 BIC R1,R1,#&20:CMP R1,#ASC"P"
400 ADRNE R0,err2:LDMNEFD R13!,{R14}
410 ORRNES PC,R14,#1<<28
420 SWI "XOS_WriteS":EQUS "Insert disc
and press any key":EQUW &D0A:EQUB 0
430 MOV R0,#15:MOV R1,#1
440 SWI "XOS_Byte":SWI "XOS_ReadC"
450 .open MOV R0,#&48:MOV R1,R12
460 SWI "XOS_Find":BVS error2
470 STR R0,[R12]
480 BL get:CMP R0,#&D:BEQ ploop
490 MOV R0,#0:SWI "XOS_Find"
500 ADR R0,err3:LDMFD R13!,{R14}
510 ORRS PC,R14,#1<<28
520 .ploop BL get:MOV R2,R0,LSL #8
530 CMP R0,#&FF:BEQ end
540 BL get:ORR R0,R0,R2
550 ADD R1,R12,#4:MOV R2,#10
560 SWI "XOS_BinaryToDecimal"
570 RSB R1,R2,#5:..s1 CMP R1,#0:BEQ s12
580 SWI &120:SUB R1,R1,#1:B s1
590 .s12 MOV R1,R2:ADD R0,R12,#4
600 SWI "XOS_WriteN":SWI &120
610 SWI "XOS_ReadEscapeState"
620 BCS end:MOV R3,#0:BL get
630 .ploop2 BL get:CMP R0,#&D
640 SWIEQ "XOS_NewLine":BEQ ploop
650 CMP R0,#&7F:SWICC "XOS_WriteC"
660 BCC ploop2:CMP R0,#&8D:BEQ 1ref
670 CMP R3,#0:SWINE "XOS_WriteC"
680 BNE ploop2:CMP R0,#&F4:MOVEQ R3,#1
690 CMP R0,#&C6:CMPNE R0,#&C7
700 CMPNE R0,#&C8:BEQ comp
710 BL tcall:B tkprt
720 .comp MOV R2,R0:BL get
730 STRB R0,[R12,#4]:ADD R12,R12,#4
740 MOV R0,R2:BL tcall:SUB R12,R12,#5
750 .tkprt LDRB R0,[R1],#1:CMP R0,#&7F
760 SWICC "XOS_WriteC":BCC tkprt
770 B ploop2
780 :
790 .1ref BL get:MOV R2,R0,LSL #2
800 BL get:AND R1,R2,#&C0
810 EOR R0,R0,R1:MOV R4,R0
820 BL get:MOV R1,R2,LSL #2
830 EOR R0,R0,R1:AND R0,R0,#&FF
840 ORR R0,R4,R0,LSL #8
850 ADD R1,R12,#4:MOV R2,#10
860 SWI "XOS_BinaryToDecimal"
870 MOV R0,R1:MOV R1,R2
880 SWI "XOS_WriteN":B ploop2
890 :
900 .get LDR R1,[R12]:SWI "XOS_BGet"
910 BVS error:MOVCC PC,R14
920 SWI "XOS_NewLine"
930 .end LDR R1,[R12]:MOV R0,#0
940 SWI "XOS_Find":LDMFD R13!,{PC}^
950 :
960 .tcall STMFD R13!,{R14}
970 ADR R14,tcall2:LDR PC,tjump
980 .tcall2 LDMFD R13!,{PC}
990 .tjump EQUD tokenaddr
1000 :
1010 .error MOV R0,#0:LDR R1,[R12]
1020 SWI "XOS_Find"
1030 .error2 LDMFD R13!,{PC}
1040 :
1050 .err1 EQUD 100
1060 EQUS "No filename":EQUB 0:ALIGN
1070 .err2 EQUD 101
1080 EQUS "Bad qualifier":EQUB 0:ALIGN
1090 .err3 EQUD 102
1100 EQUS "Not a BASIC program":EQUB 0
1110 ]NEXT
1120 SYS "OS_File",10,"BLIST",&FFC,,cod
e,O%
```

RU

TECHSOFT

DESIGNER intro

POSSIBLY THE ONLY DRAUGHTING SYSTEM YOU WILL EVER NEED

COST EFFECTIVE CAD

Created specifically for the Archimedes, **DESIGNER INTRO** is based on its big brother **DESIGNER (ARCHIMEDES).** It shares the same uniquely logical menu structure and many of the powerful draughting routines. Enough power in fact to cope with all of most peoples' draughting requirements. Many of **INTRO's** features are shown below There is one feature however that we cannot show - its most important one - the simplicity of use that guarantees a short learning curve and speedy drawing generation.

FEATURES

Fully manu driven with simple prompts and instructions for all routines.

User definable grid with grid lock.

Hunt and lock onto nearest end of line, arc or centre of a arc/circle with a single button press.

True arcs, tangents and normals, essentials for most drawing applications but particularly important for engineering applications. Nine in-built arc types alone.

Ellipses, full or part.

Powerful delete facilities, e.g., last item, line, part line, arc, part arc, etc. (The software automatically finds intersections.)

Unique MODIFY command allows drawing to be redrawn a line at a time forwards or backwards. The drawing may be 'entered' at any point thus giving total flexibility to edit and alter drawings.

Automatic or manual dimensioning, user definable arrow heads/leader lines.

Text entry at any size, angle or slope. (Text output limited only by the plotter available.)

Hatching at any angle/spacing. Automatically finds enclosed boundary plus single or multiple islands.

Powerful ELEMENT (layer) structure.

No practical limit to the number of drawing elements that may be created.

Comprehensive transformation commands, MOVE, ROTATE, MIRROR IMAGE, ALTER SIZE.

No practical limit to the size of drawing created - limited only by disk space.

Libraries of drawings can easily be created. Library items can then be incorporated into the current drawing at any size, angle, position.

Zoom, scale and pan facilities (plot-outs may be taken at any time).

Output to wide range of line plotters from A5 to A0, e.g., Graphtec, Hewlett Packard, Hitachi, Plotmate, Roland, etc. Plus screen dumps for FX/MX/RX compatible printers.

COST EFFECTIVE CAD FOR YOUR ARCHIMEDES

In creating **INTRO** TechSoft have kept the features that made **DESIGNER (BBC)** so popular. There is however one feature that we have cut back on - the cost. **INTRO** is not a toy (although you will enjoy playing with it), it is a true working system at a realistic price.

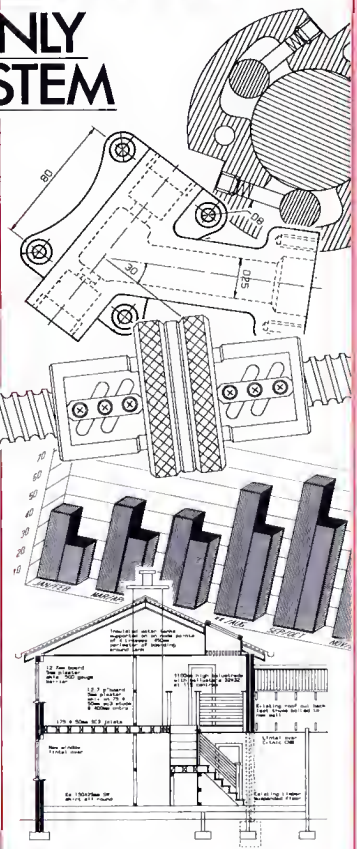
P.S. If your draughting requirements are more demanding than normal take a look at **DESIGNER (ARCHIMEDES),** more than a match for any drawing.

PERIPHERALS

Please remember that TechSoft can supply a wide range of CAD related peripherals and accessories, e.g., plotters, plotter pens, etc.

ORDERING

Cheque with order please, payable to TechSoft UK Ltd.
Add VAT @ 15% of total. (Official educational orders welcome.)



ONLY **£75** *+ VAT.*

TECHSOFT
NEW IDEAS IN SOFTWARE

TechSoft UK Ltd.
Old School Lane
Erryrys, MOLD
Clwyd CH7 4DA
Tel: 082 43 318

ARC PROCEDURE LIBRARY

by Lee Calcraft

A new occasional series, bringing you a range of useful functions and procedures. Our first library entries concentrate on mouse-related utilities.

In this series we will be bringing you a variety of procedures and functions which we hope will form a library of useful building blocks. These may be used when and where you choose, and may be incorporated into a program either by typing the code in directly, by appending it using the Arc's APPEND instruction (which automatically renumbers the incoming program lines), or by using the LIBRARY or INSTALL statements (see RISC User Volume 1 Issue 2).

Although the WIMP provides a useful environment for the Arc, it is not easy to program, and carries a significant overhead in length of code. For those who sometimes prefer to go it alone, we begin our procedure library with two mouse related utilites which do not make use of the WIMP.

PROCmousewait

This first procedure repeatedly checks the mouse and/or keyboard to see if the user has supplied an input of the required type. The type of input is specified by a single parameter, as follows:

Value	Effect
0	Wait for button release
1-7	Wait for specified button press
8	Wait for any button
16	Wait for keyboard input

For example if you insert the line:

```
PROCmousewait(4)
```

into a program, the program will halt until the Select button is pressed (since on the Arc, Select has the value 4, Menu 2 and Adjust 1). To make the procedure return after any button is pressed, use:

```
PROCmousewait(8)
```

If you add 16 to the parameter, then it will also return if the user presses any key on the keyboard. Thus:

```
PROCmousewait(1+16)
```

will wait until any key, or mouse button 1 (Adjust) is pressed, while:

```
PROCmousewait(16)
```

will wait for any key, but ignore the state of the mouse.

When the procedure returns it sets four global variables: w%, x%, y% and z%. The first takes the ASCII value of any key pressed (-1 if none pressed), while x%, y% and z% are the x and y co-ordinates of the mouse pointer at the time of selection, and z% is the value of the button pressed, or 0 if the mouse was not used.

If you call the procedure with the parameter zero, the routine will return as soon as all mouse buttons are released. This is very useful, because, even though we can flush the mouse buffer with *FX21,9, this does not prevent a user's single click from being read as a whole sequence of responses. The only way to avoid this is to wait until the mouse is clear of presses between button checks. The procedure caters for this in a second way: if you place a negative sign in front of the calling parameter, the procedure will first wait until no button is pressed, and then wait for the requested one.

FNpositiond

This very short function determines whether the mouse pointer is within a specified rectangle, and is extremely useful in the processing of mouse selections. It takes six parameters: the x and y co-ordinates of the pointer (as read by PROCmousewait for example), the bottom left-hand co-ordinates of the rectangle in question, and its width and height respectively. I have added a "d" to the function name as a reminder that the last two parameters are not absolute co-ordinates, but "differences".

As an example of the use of this function, the following two lines will wait for the Select button to be pressed, and then check if the

pointer is within a rectangle at X,Y and of width W and height H:

```
PROCmousewait(-4)
IF FNpositiond(x%,y%,X,Y,W,H) VDU7
```

If the pointer is over the defined area, a beep will sound.

The short program accompanying the two procedures in the listing provides a further example of their use. If you run it, a rectangle will appear, and each time you press the Select button, the word "Inside" or "Outside" will be displayed, indicating the position of the pointer relative to the rectangle.

Next month we will provide procedures to streamline the task of drawing selection boxes on the screen, and of determining which one has been selected by the user.

```
 10 REM             >Mselprcs
 20 REM Procedure/Function Library
 30 REM Version     A 0.3
 40 REM Author      Lee Calcraft
 50 REM RISC User April 1989
 60 REM Program     Subject to Copyright
 70 :
 80 MODE12
 90 PRINT"Select to test grid"
100 PRINT"or Menu to quit"
110 OFF:*POINTER
120 RECTANGLE 200,300,800,100
130 REPEAT
140 PROCmousewait(-8)
150 IF z%=4 THEN
160   PRINTTAB(0,6);
170   IF FNpositiond(x%,y%,200,300,800
,100) PRINT"Inside " ELSE PRINT"Outside"
180 ENDIF
190 UNTIL z%=2
200 ON
210 END
220 ================================
230 DEFPROCmousewait(n)
240 REM 0-7: Wait for this mouse state
250 REM neg: Wait for zero first
260 REM 8:   Wait for any button
270 REM 16:  Wait for keypress
280 LOCAL anypress,inkey,m
290 *FX21,9
300 IF n<0 THEN PROCmousewait(0):n=ABS
(n)
310 IF (n AND 16)>0 THEN inkey=TRUE:*F
X21
320 IF (n AND 8)>0 THEN anypress=TRUE
330 m=n AND 7:w%=-1
340 REPEAT
350   MOUSE x%,y%,z%
360   IF inkey THEN w%=INKEY(0)
370 UNTIL (z%=m AND NOT (n>7 AND m=0))
 OR (anypress AND (z%>0)) OR (inkey AND
(w%>-1))
380 ENDPROC
390 ================================
400 DEFFNpositiond(x%,y%,x1%,y1%,xd%,y
d%)
410 =x%>x1% AND x%<x1%+xd% AND y%>y1%
AND y%<y1%+yd%
420 ================================
```

RU

ADVERTISING IN **RISC USER**

For advertising details,
please contact Sarah Shrive on (0727) 40303
or FAX: (0727) 60263
or write to:

RISC USER Dolphin Place, Holywell Hill, St. Albans, Herts AL1 1EX

THE RISC OS RAM FILING SYSTEM

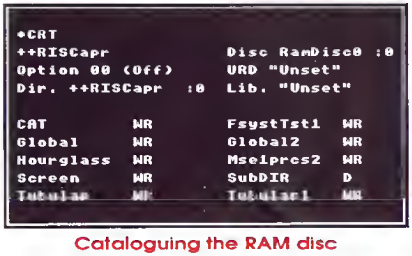
Lee Calcraft gives a brief overview of this new RISC OS feature.

On Arthur 1.2 you can configure RAM for use with the RAM filing system, but the filing system itself is not implemented. RISC OS corrects this oversight with the implementation of a full-feature RAM filing system.

To make use of it, you must first of all configure some RAM. Either use *CONFIGURE or the Task Manager from the Desktop. As an example of the former:

```
*CONFIGURE RAMFSsize 160K
```

followed by Ctrl-Break will set aside 160K of RAM for the RAM filing system. The RAM filing system icon will now appear on the Desktop icon bar alongside the floppy and or hard disc icons. Double clicking on this will open up the root directory in the same way as double clicking on any of the other drive icons.



Cataloguing the RAM disc

FROM THE DESKTOP

In fact the RAM disc will behave exactly as any floppy or hard disc. You can copy files or directories between it and any other open directory on the Desktop simply by dragging icons across in the normal way. But there are two main differences. First of all, you must remember that the contents of the RAM disc are volatile, and will be lost after a hard reset (Ctrl-Break or Ctrl-Reset or power off). But it is an easy matter to back up the RAM disc at the end of a session by dragging appropriate RAM directories across to a floppy directory on the Desktop. Secondly the RAM disc is exceptionally fast. It takes just 0.05 seconds to save an 80K screen, for example. This is 100 times faster than a floppy under RISC OS (Arthur screen saves are 300 times slower!).

One obvious application for the RAM disc is in cases where speed of loading and saving is important. For example, you could set up the RAM disc to hold eight mode 12 screens, and load them in at a speed fast enough for animation, and without wear and tear on your floppy.

OUTSIDE THE DESKTOP

If you are using your machine without the aid of the Desktop, you can again treat the RAM disc just like any other. To make the RAM filing system current, use:

```
*RAM   (*ADFS will reverse the effect)
```

You can then use *CAT to catalogue the disc, and use all the ordinary commands just as with the ADFS:

```
LOAD, SAVE *LOAD, *SAVE, *FREE, *BACK
*WIPE, *COPY, *COUNT etc etc.
```

To refer to the RAM disc when you are currently in the ADFS or Econet, use the prefix "RAM:". Thus to copy the file "MyProg" from $.Bananas on the ADFS (your current filing system) to $.Potatoes on the RAM disc, use:

```
*COPY $.Bananas.MyProg RAM:$.Potatoes.
MyProg
```

If you have a program which needs to know which is the current filing system, you can use the SYS call given in this month's Hints. It will return the value 23 for the RAM filing system.

Fans of the RISC User ADFS Menu (RISC User Volume 1 issues 1, 2 and 10) will be pleased to hear that it works well with the RAM filing system. Just use *RAM to engage the RFS. The only thing which it gets wrong is the free space display, but this is easily ignored. RU

ARE YOU GOOD ENOUGH?

As the leoders in softwore for the Archimedes ronge of computers, CLARES MICRO SUPPLIES are laaking ta extend aur ronge even further. We ore loaking for people who ore os excited by the Archimedes os we ore.

If yau hove written ony progroms, completed or not, then we would like to heor from you.

If you have any ideas far programs ond hove the obility to execute the ideos then we wont ta heor from yau.

If you hove the obility to progrom the Archimedes but nat the ideos to program then we want to heor from you.

Progroms con be written in any languoge os long os they perfarm their stated task. Mony of our programs cantain large chunks of BASIC with ARM code in the oreas thot it is needed. BASIC on the Archimedes is a very powerful longuoge and we do not ottoch ony snob volue to its use. If your progrom does what is meont to do then thots oll we are interested in. Why not join the top teom on the Archimedes. You get the suppart af aur in-house team, privileged occess through us ta Acarn and invitotions ta our informol pragrammers seminors.

The mast importont point is thot you will be eorning top royolty rates of if you prefer we will purchase your progrom autright.

Please write, in confidence, to Mr. D. Clore ot:

Clares Micro Supplies,
98 Middlewich Road,
Northwich,
CHESHIRE CW9 7DA

If you hove a progrom either complete or in development then pleose enclase o capy for aur evaluotion.

To pratect yourself we advise that yau lodge a capy of the pragram with yaur bank or solicitar BEFORE yau send us a capy. You can then prove .thot your pragrom pre-dates onything that we have.

Act tadoy ond become part af the leoding software team praducing softwore for the worlds fastest micra.

USING BASIC'S TOKEN TABLE

David Spencer shows how Basic's token table can be put to good use.

As you may be aware, when a Basic program is stored in memory (or in a file), the keywords in it (PRINT, REPEAT, SIN etc.) are replaced by a token - a number unique to that particular keyword. This saves RAM, and greatly improves execution speed because each keyword is much more easily identified when the program is run. The penalty is that the keywords must be tokenised when the program is entered, and detokenised when it is listed.

On the earlier BBC micros, any utility that wished to convert a token to its corresponding keyword, or vice-versa, either had to have a complete table of keywords (lengthy), or had to illegally access the table in the Basic ROM (prone to incompatibility problems). Basic V on the Archimedes is much more civilised, and provides a routine that converts a token into its keyword. It is this technique which is used in the program listing utility elsewhere in this issue.

The routine in question, given the name *TOKENADDR,* is one of a number of routines whose addresses are made available when a machine code routine is called using the CALL statement. These will be the subject of a future article, but all that is necessary to use the *TOKENADDR* routine is to include a short machine code patch in your program. This patch can be found in the example program (explained later) between lines 80 and 150. You should include this in your own code, and execute it when the program is first run.

To use this routine, set A% to the value of the token (and B% to the second byte for composite tokens - see below) and execute CALLcode. This will call the *TOKENADDR* routine, and then store the address where the keyword can be found in the variable *temp*. This will be a memory address within the Basic module. The keyword is terminated by a byte with a value of &7F or greater, and therefore, the keyword can be printed using the following code:

```
ptr=!temp
WHILE ?ptr<&7F
VDU ?ptr:ptr+=1
ENDWHILE
```

COMPOSITE TOKENS

The tokens for most keywords fall in the ranges &7F-&8C, &8E-&C5 or &C9-&FF. However, some of the keywords new to Basic V have two-byte tokens. These consist of a byte of &C6, &C7 or &C8, followed by a second byte identifying the actual token. The valid range for this second byte depends on the first byte:

First byte	Valid range of second byte
&C6	&8E-&8F
&C7	&8E-&9B
&C8	&8E-&A4

POINTS TO NOTE

There are a few points to note about token values:

1. The value &8D is not a valid token. Instead, it is a marker used to identify line references in GOTO, RESTORE etc. It is followed by three bytes containing a line number in coded form.

2. The keyword BY (used for relative co-ordinates) isn't tokenised

3. There are two tokens for ELSE. A value of &8C is used for an ELSE in an IF..THEN..ELSE statement (and ON..GOTO..ELSE etc.), and a value of &CC for an ELSE in an IF..THEN..ELSE..ENDIF structure.

4. TOP and LISTO do not have their own tokens as such. Instead, they use the token for TO followed by a 'P', and the token for LIST followed by an 'O' respectively. TWINO, on the other hand, does have a token separate from that for TWIN.

TOKENISING KEYWORDS

The *TOKENADDR* routine is fine for converting a token into its keyword, but what about converting a keyword into its token in the first place. Unfortunately, Basic doesn't make available the routine to do this. The only simple solution is to include a list of keywords and tokens in your program. Rather than doing this manually, you can use *TOKENADDR* with each valid token in turn to produce the entire list.

EXAMPLE PROGRAM

To allow you to experiment with the *TOKENADDR* routine, the following program will print out all the valid tokens with their corresponding keywords. Note that the composite token C8 A4 represents the keyword 'OVERLAY' which is new to the Basic 1.04 supplied with RISC OS. If this program is run on an older version of Basic, then the keyword printed for this token will be meaningless.

```
 10 REM             >TList
 20 REM Program     Basic Token Lister
 30 REM Version     A 1.0
 40 REM Author      David Spencer
 50 REM RISC User April 1989
 60 REM Program subject to copyright
 70 :
 80 DIM code 100
 90 FOR pass=0 TO 2 STEP 2
100 P%=code:[OPT pass
110 STMFD R13!,{R14}:MOV R2,R14
120 ADR R12,temp:STR R1,[R12]
130 ADR R14,back:ADD PC,R2,#&4C
140 .back STR R1,temp:LDMFD R13!,{PC}
150 .temp EQUD 0:]NEXT
160 REPEAT READ a1%,ah%,b1%,bh%
170 IF a1%<>0 THEN
180 FOR A%=a1% TO ah%
190 FOR B%=b1% TO bh%
200 CALL code
210 PRINT~A%;:IF B% THEN PRINT" ";~B%;
220 PRINTTAB(12);:ptr=!temp
230 WHILE ?ptr<&7F
240 VDU ?ptr:ptr+=1
250 ENDWHILE
260 PRINT:NEXT
270 NEXT
280 PRINT
290 ENDIF
300 UNTIL a1%=0
310 END
320 :
330 DATA 127,140,0,0
340 DATA 142,197,0,0
350 DATA 201,255,0,0
360 DATA 198,198,142,143
370 DATA 199,199,142,155
380 DATA 200,200,142,163
390 DATA 0,0,0,0
```

RU

BACKUP
IN BROMLEY

AUTHORISED ACORN DEALERS

We offer full technical backup, service and advice as well as a comprehensive range of software and hardware for your Archimedes. If you have a problem - we can help you.

Call in for a software or hardware demonstration, or phone for prices and availability

Data Store

6 Chatterton Road, Bromley, Kent
Telephone : 01-460 8991 (closed Wednesdays)

ArcEdit: the RISC OS Text Editor

Mike Williams investigates ArcEdit, the new text editor supplied with RISC OS.

Three major applications are supplied with RISC OS - Edit, Draw and Paint. In this issue of RISC User, I'll take a first look at Edit (also called ArcEdit), the text editor, and give you some thoughts on my experiences so far.

ArcEdit can be used as a simple word processor, and possesses a number of attributes which are really only relevant in that context (indent, format, choice of font, font size, line spacing and margins). However, if you intend to do much word processing, you will almost certainly find that a dedicated word processor such as 1st Word Plus or Pipedream, is essential. Formatting is very rudimentary to say the least, and there is no spelling checker. Indeed, one is almost led to wonder why any word processing features were included at all when so many are missing. But for simple word processing tasks it will suffice.

ArcEdit's main applications are likely to be in the editing of Command and Obey files (see RISC User Volume 2 Issue 4) and similar, and in editing the source code for programs in various languages (including Basic and assembler). No doubt many other needs will arise for which ArcEdit will be eminently suitable.

INSTALLING ARCEDIT

ArcEdit is a multi-tasking application, and must therefore be installed on your system each time it is switched on before you can use it. This is easily accomplished from within the Desktop by displaying the directory for the appropriate applications disc (as supplied with RISC OS), and double-clicking on the Edit icon. ArcEdit is installed in memory, and its icon appears on the menu bar at the foot of the screen.

ARCEDIT IN USE

There are three ways to start editing a document. The simplest is to click (with the select button) on the Edit icon on the menu bar, and a blank text window will appear. Alternatively, pressing the menu button while pointing to the Edit icon will display a short menu from which the Create option will generate any one of five different window types. If a file already exists, you can double click on its icon, or *drag* its icon over the Edit icon on the menu bar. The former is usually much the easier in practice.

Most if not all functions within ArcEdit can be controlled by the mouse, using the menu button (when over a text window) to display a top level menu with five options. Each of these in turn leads to a further sub-menu, and in some cases a further sideways menu choice is offered. You may find manipulating these menus tricky at first but your skill here will almost certainly improve with practice. However, many menu options are also available as direct keyboard operations.

THE MISCELLANEOUS OPTION

This provides information about ArcEdit and the current text file. A further option, called *New view*, will split the current text window into two separate windows, both on the same text file. Using this facility, different parts of the same file may be viewed at the same time, and blocks of text can be readily moved or copied from one such window to another. It would appear that you can continue to split text windows as much as you like. The only limitation is the extent to which you can manage to use such a multiplicity of windows on the screen.

THE SAVE OPTION

This is used to satisfy two slightly different requirements, to save for the first time some newly created text, and secondly to save a new copy of text previously saved. In the former case, simply type in the name you want to give the text file, display the directory in which the file is to be displayed on the screen, and drag the displayed image of the text file to the directory viewer. This can prove awkward if you have forgotten to display the relevant directory before attempting the save, or the directory has become hidden behind other windows, and the whole process is not as smooth as it might have been. However, to save a new version of

an existing file, it is only necessary to click on the 'OK' button in the save sub-menu, and this works well.

THE SELECT OPTION

This option is primarily concerned with operations on blocks of text. Once highlighted, a block can be copied, moved, deleted, saved or indented. Selected blocks can also be de-selected.

A block of text can be created by clicking the select button at the start of the block, and dragging the text cursor (a symbol referred to as a *caret*) to wherever the block ends. Unfortunately, the window does not scroll automatically if the pointer is moved below the window frame, and it thus appears impossible to mark a block which is larger than the maximum size of window (neither is there a *select-all* option). A single word can be selected by double-clicking, a whole line by triple clicking. Double clicking on a word also selects the spaces at both ends of the selected word. So using this method to delete a word leaves no gap between remaining words, and copying or moving a word will often result in an unwanted extra space.

A marked block of text may be saved to a file, and then copied to another document by positioning the pointer, and then dragging the file icon in the corresponding directory viewer onto the current text window.

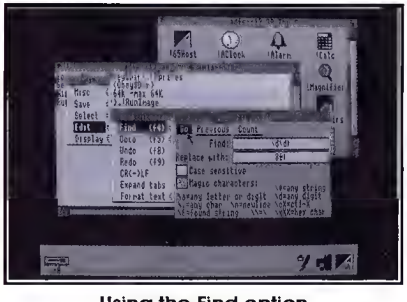
THE EDIT OPTION

The main feature of this option is a Find-and-Replace facility. Searches can be case sensitive or not, but if not, any replacement is substituted 'as is', and does not preserve any existing upper/lower case pattern. Thus a word which appears both at the start (with a capital letter) and within sentences cannot be correctly substituted for in a single search and replace operation.

The Find option can be used as a search or as a search-and-replace facility, and search and replace strings may contain a variety of so-called *Magic* characters. Thus '\d.\d' would search for a string consisting of any digit followed by '.' followed by any digit. Control codes and ASCII characters (in the range 0 to 255) may also be included.

Once a search commences, and an instance of the target string is found, the *found* box appears giving 7 different options, much as might be expected. These also include the facility to *Undo* the last change, and to *Redo* the last undone change! Although useful, there are limitations which make this feature less powerful than that provided by Edit on the BBC Master 128 for example.

A further option in the Edit menu is *Format*, allowing the column width of text to be changed. This achieves very little that is useful, and formatting of text remains largely non-existent in ArcEdit.



Using the Find option

THE DISPLAY MENU

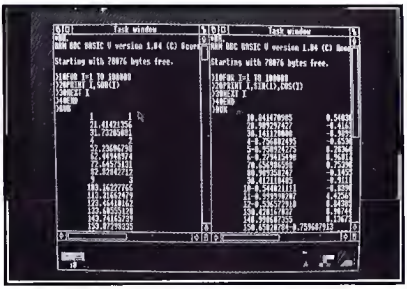
This menu provides choices which affect the way text is seen on the screen. Text font and size may be selected, but all text is formatted in the same way; there is no facility to mix fonts or sizes within the same document. Line spacing may be determined to the nearest pixel, yet only the left-hand margin may be set, and then only to the nearest character. Foreground and background colours may be selected, and text inverted if required. The wrap option, if selected, means that text is always formatted to fit the size of the window (horizontally), rather than allowing the window to scroll over the text area.

GENERAL COMMENTS

As I said at the outset, ArcEdit has many limitations when viewed as a word processor, but is adequate for the simplest of word processing tasks. It is as a text editor that ArcEdit should be used, and for this it generally performs well, but why confuse the user with extra features which are quite irrelevant in a text editor (line spacing for example)?

One particularly useful facility is the ability to select the type of file being created (by clicking the menu button when pointing to the Edit icon). Such files (Obey files for example) will then automatically be saved with the correct file type.

A further type of window is the task window. This is not an editing window, but provides an environment in which the user can initiate a new task, and has its own set of menu choices. This is a very useful feature, particularly for the more experienced user, and we shall cover the uses of this in more detail later. Suffice it to say, that by creating two such task windows, it is quite possible to have two Basic programs running concurrently (see illustration).



Using task windows to run Basic programs concurrently

These are my initial views on the new editor. Once you have your own copy installed we will be pleased to receive your own comments, and to receive any useful information for our Hints pages.

RU

ART NOUVEAU The art of the 90's

by Barry Christie

Archive (March 89)... *"Art Nouveau is a* ***superb package"***

Micronet... *"a find....its origins in Scottish education suggest that it is the art comes before the clever programming...* ***Definitely recommended"....***

ONLY £42.50 inc VAT and P&P

Order now on 24-HOUR ACCESS CREDIT HOTLINE on (0698) 733775 or send a cheque/postal order for £42.50 made payable to 'COMPUTER ASSISTED LEARNING LTD' at
Strathclyde Business Centre, Princess Road, New Stevenson ML1 4JB
Site & Local Education Authority Licenses Available.

CAL

Two spectacular programs by Mike Ironmonger.

Real-Time Image Wrapping

Last month we carried a program allowing images to be wrapped around a sphere. This was accomplished in Basic, and a sequence of sprites was created, permitting subsequent animation. This month we feature two image wrapping programs which run in real time. Both will wrap any mode 12 or 13 image. The first wraps the image around a sphere, which spins at a speed determined by the position of the mouse. The second wraps the image around a horizontal cylinder, whose speed of rotation is again mouse-controlled. The cylinder is open ended, and you can see the image wrapped onto its inner as well as its outer surface.

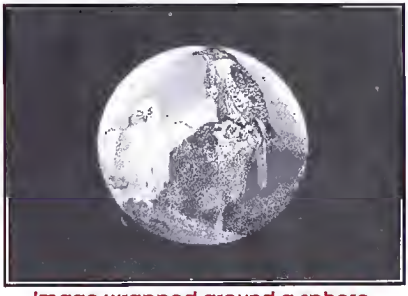


Image wrapped around a sphere

Both programs are relatively short, considering the work which they perform, and both use fully legal screen addressing code. Before running either of them, you will need an appropriate screen called "Screen" (previously saved using *ScreenSave Screen) in the current directory of your disc. The only thing to watch is that you will need 240K of screen RAM (use *Configure ScreenSize 30 on a 300 series machine). The reason for this is that three 80K screen banks are needed: one to hold the loaded screen, and two for bank switching to achieve smooth animation.

When you run the cylinder program, you can use the mouse to determine the width of the displayed cylinder. You will see two vertical bars on the screen. These represent the left and right edges of the cylinder. Moving the mouse will alter the left hand bar, while the right hand bar may be moved by pressing *Select* or *Adjust*. When you are happy with the size, press *Menu*, and the cylinder will appear.

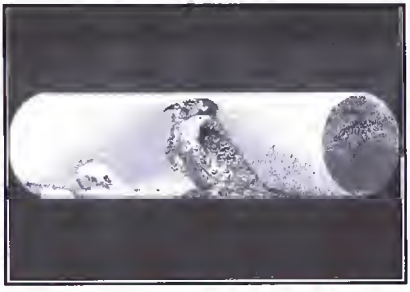


Image wrapped around a tube.
Both images are from Clare's Artisan

Notes

1. Both programs use a look-up table created in Basic to calculate the pixel positions of the wrapped image. In the case of the first program the table contains two words for every pixel that makes up the globe. The first specifies where to get the pixel from, and the second where to put the pixel. The table is a massive 170K in size, and takes some 10 seconds to generate before the image wrap can proceed. There is no noticeable delay with the cylinder program.

2. For an interesting effect, try altering line 600 of the first program to:

```
H%=320*(offset%>>3 AND 255)+X% DIV 4
```

3. The appearance of the tube in the second program can be altered by adjusting the values

of the two variables in line 500. They determine the horizontal (H%) and vertical (V%) radius (in pixels) of the mouth of the tube. If H% is negative, then the mouth of the tube appears at the left instead of the right. Note also that you should keep V% to 43 or less.

```
   10 REM            >Tubular
   20 REM Program    Rotating screen
   30 REM Version    A 1.00
   40 REM Author     Mike Ironmonger
   50 REM RISC User April 1989
   60 REM Program    Subject To Copyright
   70 :
   80 ON ERROR MODE0:PRINT REPORT$;" at
line ";ERL:END
   90 MODE0:DIM code% 2000,addr%(3)
  100 :
  110 FORA%=0TO1:P%=code%
  120 [OPT A%*2
  130 ADR    R8,table
  140 MOV    R9,#256
  150 .next_line
  160 LDR    R0,[R8],#4
  170 TEQ    R0,#0
  180 BEQ    no_line
  190 ADD    R10,R7,R5
  200 ADD    R11,R0,R6
  210 MOV    R3,R4
  220 .next_byte
  230 LDRB   R0,[R10],#1
  240 STRB   R0,[R11],#1
  250 SUBS   R3,R3,#2
  260 LDRNEB R0,[R10],#1
  270 STRNEB R0,[R11],#1
  280 BPL    next_byte
  290 .no_line
  300 SUBS   R5,R5,#320
  310 ADDMI  R5,R5,#81920
  320 SUBS   R9,R9,#1
  330 BNE    next_line
  340 MOV    PC,R14
  350 :
  360 .list EQUD150:EQUD-1:EQUD0
  370 .table
  380 ]:NEXT
  390 :
  400 SYS "OS_ReadVduVariables",list,lis
t+8
  410 IF list!8<240*1024 PRINT"240K of s
creen memory required":END
  420 *ScreenLoad Screen
  430 OFF:IF MODE<>12 AND MODE<>13 PRINT
"MODE 10,12,13 only":END
  440 :
  450 !list=148:FOR A%=3 TO 1 STEP-1
  460 SYS "OS_Byte",112,A%:IF A%>1 CLS
  470 SYS "OS_ReadVduVariables",list,lis
t+8
  480 addr%(A%)=list!8:NEXT
  490 :
  500 H%=32:V%=43
  510 FOR A=0 TO PI*2 STEP PI/128
  520 OA%=A%:X%=H%*SIN(A)+.5:Y%=V%*COS(A
)+.5
  530 A%=Y%*320-X%:!P%=A%*(A%<>OA%)
  540 P%+=4:NEXT
  550 :
  560 GCOL 3,1:MOUSE TO 480,0
  570 OX%=480:OW%=316:PROCrec
  580 REPEAT
  590  W%=OW%+4*(B%=4 AND OW%>4)-4*(B%=1
 AND OW%<1280-8*ABS(H%))
  600  MOUSE RECTANGLE 0,0,1276-W%,0
  610  MOUSE X%,Y%,B%
  620  WAIT:PROCrec:OX%=X%:OW%=W%:PROCre
c
  630 UNTIL B%=2:PROCrec
  640 :
  650 *Pointer
  660 MOUSE TO 640,512
  670 E%=OW%/4:H%=addr%(1)+OX%/4
  680 addr%()+=(159-OW%/8+320*127)
  690 offset%=0:bank%=2:*FX 113,3
  700 REPEAT
  710  MOUSE X%,Y%,B%
  720  offset%+=Y%DIV4-128
  730  F%=320*(offset%>>3 AND 255)
  740  G%=addr%(bank%):CALL code%
  750  WAIT:SYS "OS_Byte",113,bank% TO ,
bank%
  760 UNTIL FALSE
  770 :
  780 DEFPROCrec
  790 RECTANGLE OX%,-4,OW%,1028
  800 ENDPROC
```

We offer a unique inspection copy service

NOW AVAILABLE

Call for full details

ARCHWAY

Have you spent long hours struggling with the programming complexities of the brilliant Archimedes **WIMPS** system?

With **ARCHWAY** that's all behind you. Now YOU can easily and quickly build multi-window programs with pop-up menus, icons, mouse control, etc. of a professional quality.

ARCHWAY provides tools, bricks, mortar and a basic structure. YOU slot the final bricks into place and add finishing touches.

A comprehensive suite of integrated tools lets you define and edit windows, menus, icons, dialogue boxes, mouse pointers, sprites, fill and line patterns, anti-aliased fonts, short cut keys and much more! You can import files (eg ARM machine code and music editor).

The bricks include an extensive library of functions and procedures providing ready access to many of Archimedes' most powerful features.

An in-depth user guide (300+ pages, ring bound) takes you gently step-by-step through a progressive series of program building examples with the emphasis on clarity.

The ARCHWAY run-time package together with script shells in BBC BASIC provide the structure.

£79.95 incl. VAT & p/p

The complete system comes on 4*800k discs (3 tools & 1 run-time). You need 1M of RAM to run the tools. One disc drive is sufficient.

SPECIAL INTRODUCTORY OFFER: No charge for upgrade to the extended RISC OS 2.0 version we will release in May 1989. (£99.95)

SIMTRON *Programs to help you*

4 CLARENCE DRIVE, EAST GRINSTEAD, WEST SUSSEX RH19 4RZ Telephone (0342) 328188

Access

Cheques/POs/official orders or Access/Visa number and expiry date. 24-hour 'phone for Credit Card orders.

VISA

```
   10 REM             >Global
   20 REM Program     Spinning globe
   30 REM Version     A 2.00
   40 REM Author      Mike Ironmonger
   50 REM RISC User April 1989
   60 REM Program     Subject to copyright
   70 :
   80 ON ERROR MODE0:REPORT:PRINT" at li
ne " ;ERL:END
   90 :
  100 DIM screen%(3),sin(128)
  110 DIM code% 1000,table% 21736*8
  120 :
  130 FOR A%=0 TO 1:P%=code%
  140 [OPT A%*2
  150 LDR   R8,pixs
  160 ADR   R9,table%
  170 .next_block
  180 ]
  190 FOR B%=1 TO 19
  200 [OPT A%*2
  210 LDMIA R9!,{R1-R4}
  220 SUBS  R1,R1,R7
  230 ADDMI R1,R1,#81920
  240 LDRB  R0,[R5,R1]
  250 STRB  R0,[R6,R2]
  260 SUBS  R3,R3,R7
  270 ADDMI R3,R3,#81920
  280 LDRB  R0,[R5,R3]
  290 STRB  R0,[R6,R4]
  300 ]:NEXT
  310 :
  320 [OPT A%*2
  330 SUBS  R8,R8,#38
  340 BNE   next_block
  350 MOV   PC,R14
  360 .pixs EQUD 21736
  370 .in   EQUD 150:EQUD -1
  380 .out  EQUD 0
  390 ]:NEXT
  400 :
  410 SYS "OS_ReadVduVariables",in,out
  420 IF !out<&3C000 PRINT"Program requi
res 240K of screen memory":END
  430 *ScreenLoad Screen
  440 IF MODE<>12 AND MODE<>13 PRINT"MOD
E 12 or 13 only":END
  450 !in=148:FOR A%=1 TO 3
  460 SYS "OS_Byte",112,A%:IF A%>1 CLS
  470 SYS "OS_ReadVduVariables",in,out
  480 screen%(A%)=!out
  490 NEXT:OFF
  500 :
  510 PROCcreate_table
  520 *FX 113,3
  530 *Pointer
  540 MOUSE TO 640,512
  550 F%=screen%(1):offset%=0:bank%=2
  560 REPEAT
  570  MOUSE X%,Y%,B%
  580  offset%+=128-Y%DIV4
  590  G%=screen%(bank%)
  600  H%=320*(offset%>>3 AND 255)
  610  CALL code%
  620  WAIT:SYS "OS_Byte",113,bank% TO ,
bank%
  630 UNTIL FALSE
  640 :
  650 DEFPROCcreate_table
  660 A%=0:FOR A=0 TO PI/2 STEP PI/256
  670 sin(A%)=SINA:A%+=1:NEXT
  680 R%=83:OX%=1:x%=0:T%=table%
  690 FORX=0 TO PI/2 STEP PI/320
  700 X%=.5+R%*SINX:IF X%<>OX% PROCline
  710 x%+=1:NEXT
  720 ENDPROC
  730 :
  740 DEFPROCline
  750 H%=R%*COSX:OX%=X%:OY%=1
  760 FOR A%=0 TO 127:Y%=H%*sin(A%)
  770 IF Y%<>OY% PROCpix
  780 NEXT:ENDPROC
  790 :
  800 DEFPROCpix
  810 T%!0=41120+A%*320+x%
  820 T%!4=41120+Y%*320+X%
  830 T%!8=41119+A%*320-x%
  840 T%!12=41119+Y%*320-X%
  850 T%!16=40800-A%*320+x%
  860 T%!20=40800-Y%*320+X%
  870 T%!24=40799-A%*320-x%
  880 T%!28=40799-Y%*320-X%
  890 OY%=Y%:T%+=32
  900 ENDPROC
```

RU

COMPUTER CONCEPT'S SCANNER

David Spencer looks at Scan-Light, Computer Concept's new scanner package.

Scan-Light from Computer Concepts is a system that takes a printed image and reads it into a sprite. This is an integral part of Computer's Concepts Fax system, and the company has decided to release the scanner as a separate product.

SCAN-LIGHT

Computer Concepts' package consists of a hand-held Mitsubishi scanner, a podule which fits inside the Archimedes, and a user guide. All the software to control the scanner is included in an EPROM on the podule, and therefore no disc-based software is supplied. Obviously, a podule back-plane is necessary to use Scan-Light.

The scanner podule at first sight appears to be a largely unpopulated circuit board. This is because the board is actually a cut down version of the long-awaited Fax podule. CC supply a half width blanking plate to allow the podule to be mounted properly, and installation is a fairly simple process. The podule back-panel carries a socket for connecting the scanner, and a couple of unused holes, again a carry over from the Fax system.

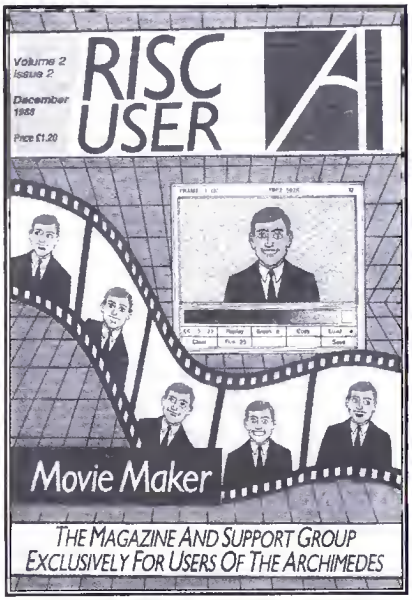
When the computer is switched on, the presence of the scanner podule is not announced in any way. It is only when you type *SCAN, or use the RISC OS task manager to start up a task called 'Scan', that the system springs into life. Scan-Light is fully compatible with both Arthur 1.20 and RISC OS, though I shall only deal with RISC OS here. Once started, an icon appears on the icon bar, and a large window is opened that will eventually display the scanned image. This window is peculiar, because clicking on the close box clears the work area of the window, rather than simply closing it. This seems to make the iconbar icon somewhat redundant.

Another novel feature is the inclusion of a menu bar. The software uses a two- level menu structure which can be invoked using the menu button as normal. The purpose of the menu bar is to bypass the first level of the menu tree, which can make the selection of functions much quicker. Keyboard shortcuts exist for many of the menu options. There are three main menus: File, Scan and Edit.

The Scan-Light system

Perhaps the most important item on the menu is Scan. This offers two options, the first to perform a scan, and the second to power down the computer. This latter is another feature aimed at the Fax system, and will only work with a special power switching unit. Once the scan option has been selected, the software waits until the button on the scanner is pressed. The scanner should then be pulled down the page being scanned. The scanning stops when the button is released, although it will also stop when a full A4 page has been read in, or if the scanner is not moved for a couple of seconds or more. A roller on the underside of the scanner detects the speed of movement, and therefore it is not essential to move the scanner at a constant rate. There are limits, however, and a page cannot be scanned in less than about nine seconds. It is also essential not to move the scanner across the page as it is moved, otherwise the image will be distorted. To achieve this, the scanner has graduation marks to line up with the page.

There are two modes of scanning offered - Picture and Text. The text mode creates a simple black and white image of the page at a resolution of 200 dots per inch. The picture mode on the other hand uses a dithering technique to create the appearance of sixteen grey levels. This method, as its names suggests, is best suited to the scanning of pictures and photos.



A scanned image

As the page is scanned, the image is built up in the window, although the scaling applied means that it will look a little strange. Once scanning is completed, the image can be viewed at 25, 50, 100 or 200% scales by selecting options from the Edit menu. A further option allows a scanned picture to be displayed in sixteen grey levels. The scanner assumes that pixels are square in shape, and therefore, scanned images will appear best in mode 9, or ideally mode 20, but this latter requires a multi-sync monitor. Using mode 12 or 15 gives the image an elongated appearance on screen. Other Edit options allow you to edit the image pixel by pixel, and to select an area of the image for future operations.

The File menu allows images to be loaded and saved, and sprites to be included within the image. Rather than follow the standard RISC OS Filer system, Computer Concepts has decided to design their own dialogue box for file operations. In use, I found this was more restricting than having Directory Viewers, and I don't fully agree with CC's departure from the standard which Acorn is eager for developers to follow. Images can be stored in one of two formats. The first is as a normal sprite, in which case the dithered effect is converted to grey levels, and the second as a compressed format which will be compatible with the Fax system. The first method has the advantage that a standard sprite file is produced, so this can subsequently be loaded into ArcPaint and edited, or incorporated into word processor documents etc. The second method saves disc space (an A4 sprite takes up a massive 483K - compression saves about 50%), but for all intents and purposes is useless without the Fax system.

DOCUMENTATION

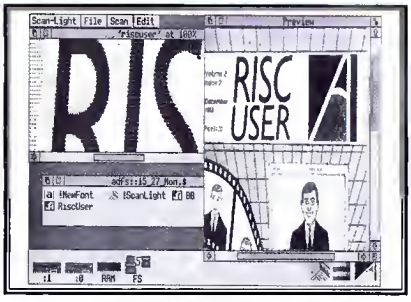
The Scan-Light package reviewed was a pre-release version, and the documentation was in fact the complete manual for the Fax system. However, we understand that the Scan-Light manual will consist of the relevant parts of this, and should end up as a 30-40 page A5 book, which will most likely be wire bound in CC's normal style.

The manual covers the principles behind the scanner, and explains in detail how to scan documents and use the other features of the software. The manual also warns that occasionally the scanner produces slight errors in the image (a fault of the scanner unit), and that red printing is not picked up by the scanner. The reason for this is that the scanner works by bouncing red light off the page as it is

scanned. This might sound a bit limiting, but in practice is not too great a problem because much apparently red printing actually contains a fair degree of black, and a brightness control on the scanner allows you to increase the sensitivity to pick this up.

SHEET FEEDER

An optional sheet feeder is also available. The scanner clips into this, and can then scan a page fed through automatically. This ensures that the scanner is not moved side to side as the page is scanned. The Scan-Light software includes all the support needed for the sheet feeder, and the unit simply plugs in between the scanner and the computer.



Screen display

CONCLUSION

Scan-Light is certainly a very addictive tool, and for the right people it will also have many serious uses. In particular, when desktop publishing packages become available for the Archimedes, it will be a simple matter to include scanned pictures on a page. However, Scan-Light costs £516.35 (inc. VAT), and I feel that this will put many people off. The optional sheet feeder adds another £172.50 to the price, giving a total of nearly £700. This is comparable with a video digitiser and camera - a setup that could in some cases perform a similar function to a scanner. As nice as it is, Scan-Light is very much for the rich Archimedes owner.

RU

Archimedes Shareware Discs

Graphics Demo 1 (£3.50)

49 graphics demo programs plus MenuMaster – a program to allow you to edit the demo order and add your own favourite demos.

Shareware Disc Nº 1 (£3.50)

MenuMaster with seven more graphics demo programs plus Life, Mandelbrots, Structured Directory Lister, European Geography.

Shareware Disc Nº 2 (£4.50)

DFS reader, backup and archiver. 9 Graphics demos. Connect four, Mastermind, Solitaire and Star Trek games. 256 colour Sprite Editor. CMOS ram Editor, Disc Editor, LQ printer font definer, Matrix Functions, Memory Mappings & Vector Listings BASIC fast Screenload.

Save £1.50 – Buy all 3 for just £10.

Also Available...

WIMP Template Editor (£8)

Design your own windows, icons, menus and sprites – completely freehand in a wysiwyg format. Create WIMP templates which can easily be loaded from your own program – no more need for endless data blocks!!!!

Based on the original "Form Editor" program from Acorn Computers, modified and extended (with Acorn's permission!)

Prices include UK p&p. European prices as UK. Outside Europe sent by surface but add £1 per order for airmail. Pay by UK cheque or Eurocheque or credit our Girobank account 2341107. Sorry no credit cards.

Norwich Computer Services, 18 Mile End Road, Norwich, NR4 7QY. (0603–507057)

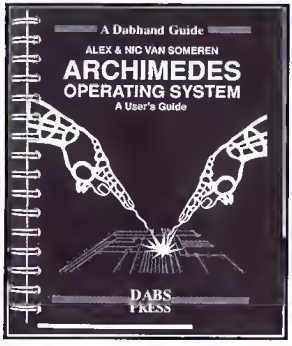
ARCHIMEDES OPERATING SYSTEM (A USER'S GUIDE)

Reviewed by David Pilling

The natural reaction to a new book on the Archimedes OS, is Yippee! I can throw away my copy of Acorn's Programmer's Reference Manual (PRM). But as we shall see, this is not possible just yet.

Title	Archimedes Operating System (A User's Guide)
Authors	Alex and Nic van Someren
Publisher	Dabs Press
Price	£14.95.

In fact, Dabs new book (which I'll refer to by its initials AOS) falls into the area between the Archimedes User Guide and the PRM. This is difficult territory, since there is a danger of just re-hashing Acorn's User Guide for the Archimedes. The book actually opens by admitting that a copy of the PRM will also be needed by those producing serious applications.



AOS is aimed at people who have read the User Guide and would like to move on to something more advanced, or those who have bought the PRM and would like some further explanation. The reader is assumed to have just the standard system: no space is devoted to things like the Acorn linker, or the various compilers etc. now available. Interfaces to the OS are described from either Basic or Basic assembler. In contrast, despite the *A User's Guide* subtitle, AOS is really of more use to programmers than users. In fact non-programmers would probably be best advised to avoid this book.

One problem the writers have had to face is the now imminent release of RISC OS. Is the book compatible with this? In fact, the minimum of revision has been undertaken to ensure that all the features of Arthur described are relevant under RISC OS. So you can happily use the book with RISC OS, but some of the new features which are very useful even at a basic level, like the printer drivers and draw module, are omitted.

Having said that, coverage is pretty comprehensive: there are chapters on everything from the philosophy behind RISC processors to how to write your own modules and applications, including assembly language and the ARM chip set. As one might expect, there is information on the usual SWI calls concerned with files, miscellaneous OS functions, and character input and output. Other chapters cover vectors and interrupts, and conversion SWI calls. There are four chapters on the sound system as well as a chapter each on the Window Manager and font system.

There is a good description of the command line interpreter, both from the star prompt, and from Basic or assembler (using OSCLI). This deals with such subjects as aliases and OS variables. Another sizable chunk deals with filing systems, the file switch module and filing system SWI calls.

The chapters on modules and applications give coverage to the neglected subjects of memory management including claiming memory from the heap. A demonstration 'shell' program shows how to run one application from inside another.

The description of the support chips (like VIDC and MEMC) is interesting, but probably stops just short of being really useful. The programmer's interface to the floating point unit is also described.

Such comprehensiveness leads one to wonder whether a mere 14 pages on the Window Manager will convert the novice into a WIMP expert, or if the space could have been better used. Given this constraint, the chapter on the Window Manger is not too bad, with an extensive example at the end. However, not all the Arthur 1.2 features are covered, and even the RISC OS extensions to the WIMP which a beginner could use are not mentioned.

The chapters on the sound system present this difficult subject in a clear way. They

culminate in a voice generator module which will convert a sampled waveform into a voice module.

The style of the book throughout is to take a section of the OS, describe it in general terms, investigate the various ways of calling the routines in it, and then provide some tutorial programs along with a few hints and tips. The programs are for the most part fairly illuminating and useful (e.g. a printer buffer module) but not exciting. A disc containing them is available for £9.95, or the book and disc can be obtained together for £21.95. There could have been more description of what the snags are with the various types of programs. For example, WIMP programs usually suffer from one set of problems, modules from another.

Most of the technical information in the book can be found in either the PRM or the User Guide. There are interesting snippets throughout, and of course all of the examples are new. Some things which would take great powers of deduction to derive from the PRM are brought into the light. The contentious issue of the serial port is for the most part avoided. This is understandable, but many people (programmers and users alike) will have to use this at some time, even if they aren't interested in communications as such. A few notes on how to wire serial leads would have been most useful.

The authors write in an informal and interesting way, which makes the book easy to read, and although I can believe that it would be easier to find information more quickly in this book than from the PRM, neither approaches the quality of the old BBC Advanced User Guide as a high speed reference work.

AOS is spiral bound with a handy place marker in the cover, and is highly professional in appearance. Archimedes books are hardly commonplace, and Dabs are to be congratulated for putting so much effort into something for the Archie.

Overall, a well written and well presented book offering good value for money. Those looking for the whole truth about the Archie OS will have to wait for a far weightier (and probably more expensive) tome in the future. For many people though, this will make interesting and useful reading for now. RU

DIARY for Archimedes

See a full year at a glance.
A single screen shows part of the entry for every day of the year. See the free days or spot that elusive entry at a glance.

Instant viewing of the full entry for any day.
Skim the mouse over the year to scan the full entries for any day, & just 'click' on a day to make an entry.

Print all of a month with just one command.
Select the month, or months, to be printed. Then choose from either a full page, weekly formatted print-out, or a quick single line per day, showing only those days which have an entry.

"Arthur" version £9.95
(incl vat & U K p&p)

MULTI-TASKING "RISC-OS" DESKTOP version £14.95
Ready in April will allow user selection of larger diary sizes & string search facility.

****** SPECIAL OFFER for this month ******
"Arthur" version including upgrade, only £10.95

Cheque or P.O. to Official Orders also welcome
QUERCUS COMPUTER SYSTEMS
Courtyard House, North Otterington,
Northallerton, N.Yorks. DL7 9EP (tel. 0609 7064?)

GEOSCAN

A NEW GEOGRAPHY AND ECONOMICS PROGRAM ON WORLD DEVELOPMENT

WRITTEN SPECIALLY FOR THE ARCHIMEDES SERIES 310 & ABOVE

Simon Beaton B.Sc & Ron Dutton Ph.D

GEOSCAN consists of a single disc containing a large and varied database covering locational, physical, climatic and socio-economic information on 100 countries of the world. With this data there are program facilities for displaying the information in the form of text, statistics, graphs and diagrams, as well as for analysing and relating the data through rankings, correlations, scattergraphs, averages and mapped locations.

GEOSCAN has wide applicability as a straightforward and self-contained learning resource which will give students of all ages, valuable experience in the use of the computer for information retrieval. However, the comprehensive ASSIGNMENT PACK which accompanies the program has been written especially for students working at the level of GCSE and beyond. The content and the ready-to-duplicate nature of the worksheets will make them appeal particularly to teachers seeking Assignments for GCSE Geography and Economics.

An important feature of GEOSCAN is the annual up-date of its database as the latest statistics become available from international agencies. Each year, a new disc will be offered at very low cost, to the registered holders of the program , in exchange for their existing disc.

PRICE:£45.00 including VAT

Order direct, or send for information leaflet to:

PASSKEY MARKETING
P.O. Box 649 Shenley Lodge Milton Keynes MK5 7AX

```
 2330 F1MUL=F1XSUB+4:F1XDIV=F1MUL+4
 2340 FLOATB=F1XDIV+4:FINTEGB=FLOATB+4:F
SQRT=FINTEGB+4
 2350 PROCB(F1LDA):PROCB(F1MUL):PROCB(F1
ADD)
 2360 PROCB(F1XSUB):PROCB(FSQRT):PROCB(F
1STA)
 2370 FACC=0:FACCX=1:FGRD=2:FSIGN=3:SP=1
3
 2380 REM Call with CALL CODE%,c(),sz(),
b(),H%,L,M%
 2390 REM c array, Sz array, b array, H%
 hit/nohit, L length, n which sphere
 2400 ADDRn=0:ADDRL=ADDRn+8:ADDRH=ADDRL+
8:ADDRb=ADDRH+8:ADDRSz=ADDRb+8:ADDRc=ADD
RSz+8
 2410 FORZ=0TO2STEP2
 2420 P%=CODE%
 2430 [OPT Z
 2440 STMFD SP !,{R0,R14}
 2450 MOV R10,R9 ;arg 1ist
 2460 MOV R11,#0 ;N%
 2470 LDR R12,[R10,#ADDRb] ;b()
 2480 LDR R12,[R12]
 2490 ADD R12,R12,#12 ;base of b()
 2500 .LOOP ADD R9,R11,R11,LSL #2
 2510 ADD R9,R9,R12 ;address% of b(N%)
 2520 BL F1LDA
 2530 BL F1MUL
 2540 LDR R7,[R10,#ADDRSz]
 2550 LDR R7,[R7]
 2560 ADD R7,R7,#12
 2570 ADD R6,R11,R11,LSL #2
 2580 ADD R9,R6,R7
 2590 BL F1ADD
 2600 LDR R7,[R10,#ADDRc]
 2610 LDR R7,[R7]
 2620 ADD R7,R7,#12
 2630 ADD R6,R11,R11,LSL #2
 2640 ADD R9,R6,R7
 2650 BL F1XSUB
 2660 TEQ FSIGN,#0
 2670 BPL LOOPEND
 2680 MOV FSIGN,#0
 2690 BL FSQRT
 2700 ADD R6,R11,R11,LSL #2
 2710 ADD R9,R6,R12 ;address% of b(N%)
 2720 BL F1ADD
 2730 TEQ FACC,#0
 2740 EORNE FSIGN,FSIGN,#&80000000
 2750 TEQ FSIGN,#0
 2760 BMI LOOPEND
 2770 CMP FACCX,#&70
 2780 BCC LOOPEND ;if t<1e-5 break
 2790 LDR R7,[R10,#ADDRH]
 2800 LDR R6,[R7]
 2810 CMP R6,#0
 2820 BEQ NOHIT
 2830 MOV R6,#0 ;if H%
 2840 STR R6,[R7] ;H%=0
 2850 LDR R9,[R10,#ADDRL]
 2860 BL F1STA ;L=t
 2870 STR R11,[SP] ;M%=N%
 2880 B LOOPEND
 2890 .NOHIT STMFD SP !,{R0,R1,R2,R3} ;e
lse
 2900 LDR R9,[R10,#ADDRL]
 2910 BL F1LDA
 2920 LDMFD SP !,{R4,R5,R6,R7}
 2930 BL FCMP ;test L>t
 2940 BCC LOOPEND ;ift<L
 2950 MOV R0,R4
 2960 MOV R1,R5
 2970 MOV R2,R6
 2980 MOV R3,R7
 2990 BL F1STA ;L=t
 3000 STR R11,[SP] ;M%=N%
 3010 .LOOPEND ADD R11,R11,#1
 3020 LDR R7,[R12,#-12]
 3030 CMP R7,R11
 3040 BCS LOOP
 3050 LDMFD SP !,{R0}
 3060 LDR R9,[R10,#ADDRn]
 3070 STR R0,[R9]
 3080 CMP R0,R0
 3090 LDMFD SP !,{PC}
 3100 .FCMP CMP (FSIGN+4),FSIGN ;check r
0-r3 against r4-r7
 3110 BNE FCMPDONE
 3120 TEQ FSIGN,#0
 3130 BMI FCMPNE
 3140 CMP FACCX,(FACCX+4)
 3150 CMPEQ FACC,(FACC+4)
 3160 MOV PC,R14
 3170 .FCMPNE CMP (FACCX+4),FACCX
 3180 CMPEQ (FACC+4),FACC
 3190 .FCMPDONE MOV PC,R14
 3200 ]
 3210 NEXT
 3220 ENDPROC
 3230 :
 3240 DEF PROCB(RETURN A)
 3250 A=A+8+(!A AND&FFFFFF)*4
 3260 ENDPROC
```

PIPEDREAM

POWER AT YOUR FINGERTIPS

PIPEDREAM

- The best word processor for the Archimedes. It has dynamic on-screen page breaks, multi-column text and 'live' embedded numbers and calculations.
- The fastest and most powerful spreadsheet for the Archimedes. It can sort, format text around numbers and store and quickly recalculate large models
- The most flexible database for the Archimedes. It can store a huge number of fields, search, sort and select records and have fields which are calculated from other values.
- Complete integration of word processor, spreadsheet and database – no need to exchange data between several programs.
- A mail-shot program which can store names and addresses, write the letter to send to them and selectively mail to each one, including calculations.
- Ideal for writing invoices, statements and estimates: the text and figures can fit easily into a pre-designed form – and, of course, the figures are 'live'.
- An excellent way to record customer or sales information: keep a database of sales leads, sort it on different criteria, produce the required totals and select and print out the key information.
- Compatibility with 4 different computers: the BBC Micro, the Acorn Archimedes, the Z88 and the IBM PC.
- A pull-down menu and help system to make it easy for you to get started.
- Built-in file transfer with PipeDream on the portable Z88 so that you can continue work away from your main computer.

SPELLCHECK

- PipeDream SpellCheck is now available, with a dictionary of 93,003 words and the ability to check over 40,000 words per minute on the Archimedes.

VIEW PROFESSIONAL

- The BBC Micro version of PipeDream which provides complete file compatibility with PipeDream and with the Z88 and which shares many of PipeDream's features.

PipeDream costs only £99.00 plus VAT, View Professional £60.00 plus VAT and SpellCheck £43.00 plus VAT

To order PipeDream, View Professional or SpellCheck, or to find out more, call us on 0954 211472. You can pay by VISA or Access by phoning us with your card number.

colton software

FOR A FREE BROCHURE, COMPLETE AND SEND THIS COUPON

PipeDream ☐ *View Professional* ☐ RU3/89

Name

Address

Post Code

COLTON SOFTWARE, BROADWAY HOUSE,
149-151 ST. NEOTS ROAD, HARDWICK,
CAMBRIDGE CB3 7QJ, ENGLAND
FAX (0954) 211607

Using the RISC User Toolbox with RISC OS

David Spencer updates the RISC User Toolbox to work with RISC OS.

Since details of RISC OS became available, we have endeavoured to make all our programs fully compatible with it (as well as with Arthur). However, the *RISC User Toolbox* was started well before RISC OS was announced, and is not fully compatible as it stands. Most of the commands do however function correctly, there being only two problems. Firstly, the disassembler incorrectly shows the destination of branch instructions, and secondly, the command *DEDITF (invoke the disc editor at the start of a named file) does not work. Adding the listing given here to the complete Toolbox source program resolves the first problem, and partly cures the second. The amendments also change the title of the module to 'Toolbox' (instead of 'RiscUserToolbox'). This allows *HELP TOOLBOX to be used.

E-FORMAT DISCS

The 'E' format for discs uses a scatter map technique to allow files to be split in several chunks. Because of this, the *DEDITF command, will not work with 'E' format discs, even with the amendments included (although no harm we result if you try it). The other disc editor commands, DEDIT and DEDITT, work as before. However, care should be exercised when using any disc editor on an 'E' format disc, as the layout of these discs is more complex than that of the other formats. This, coupled with the fact that files are no longer stored in a single contiguous block, makes it all too easy to accidentally destroy a 'sensitive' area of the disc, and possibly render that disc totally unrecoverable.

```
   170 EQUS "Toolbox":EQUB 0
  6025 MOV R5,R1:MOV R1,#1:MOV R0,#15
  6026 SWI "OS_Byte":MOV R1,R5
 10470 MOV R0,#32:ADD R1,R12,#1024
 10570 ADD R1,R1,#64
 10605 SWI &117:SWI &110
 10606 MOV R0,#8:BL sndnull
 10880 .swi8 STMFD R13!,{R14}
 10890 SWI "Debugger_Disassemble"
 10900 LDMFD R13!,{PC}
```

Delete lines 10910 to 11060.

RU

ACORN'S RAY TRACER (continued from page 47)

```
 3270 :
 3280 DEF PROCTMDSAVE(A$)
 3290 SYS "OS_SpriteOp",SO_Save,SA%,A$
 3300 ENDPROC
 3310 :
 3320 DEF PROCTMDLOAD(A$)
 3330 SYS "OS_SpriteOp",SO_Load,SA%,A$
 3340 ENDPROC
 3350 :
 3360 DEF PROCHDR
 3370 CLS
 3380 PRINT"Computing frame ";frame
 3390 PRINT"Y coordinate=";:cx%=POS:cy%=
VPOS
 3400 ENDPROC
 3410 :
 3420 DEF PROCinit
 3430 lsl(0)=-30:lsl(1)=20:lsl(2)=-30
 3440 lsl()=lsl()/MODlsl()
 3450 margin=0:maxframe=129
 3460 width=64:height=64
 3470 LM=0+margin:RM=width*4-1-margin
 3480 BM=0+margin:TM=height*4-1-margin
 3490 XC=width*2:YC=height*2:VD=3800
 3500 NoSpheres=15
 3510 SASIZE%=130*(width*height+256)
 3520 res=1
 3530 ENDPROC
```

Listing 2

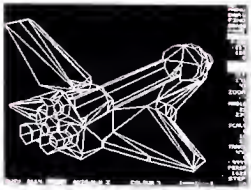
```
   10 REM >Display
   20 REM Display ray traced sequence
   30 X%=OPENIN"RaySprites":T%=EXT#X%+16
   40 CLOSE#X%:DIM sp T%:!sp=T%:sp!4=0
   50 sp!8=16:sp!12=16
   60 SYS "OS_SpriteOp",&10A,sp,"RaySpri
tes"
   70 MODE 13:OFF
   80 REPEAT
   90 FOR frame=0 TO 129
  100 SYS "OS_SpriteOp",&122,sp,RIGHT$("
000"+STR$frame,3),512,384
  110 WAIT:WAIT
  120 NEXT
  130 UNTIL FALSE
```

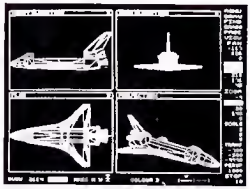
RU

SILICON VISION

SOFTWARE FOR THE ARCHIMEDES & BBC

SolidCAD





The ultimate 3D Draughting System for Architectural design, Interior design, Engineering Design and Teaching CDT. Allows drawing in plan, front & side elevations and also directly in 3D view. Includes powerful zoom & pan options for precision draughting and surface definition for creating solid colour objects. Also includes Sweep, Extrude & Macro facilities for designing very complex objects easily. Designs created with SolidCAD are compatible with the Realtime Graphics Language for high-speed flicker-free animation. The custom Archimedes version also performs smooth shading for realism. SolidCAD(Arc) users can upgrade to the Realtime Solids Modeller (Arc) for £40.00.

£49.95 (ARC & BBC) New

REALTIME SOLIDS MODELLER

The package includes both the sophisticated design environment of SolidCAD and the high speed animation capability of a Realtime Graphics Language (RGL) module developed in pure ARM Risc code for supercharged performance. The package is ideal for Architectural design, Interior design, Engineering design & teaching CDT. The RGL module can be used to create standalone flicker free animation of designs from your own programs. Smooth shading is also performed for realistic images. Through our in-house expertise in 3D Design and High-speed techniques, no other package can rival the design environment & animation speed of the Realtime Solids Modeller.

£89.95 (ARC) New

REALTIME GRAPHICS LANGUAGE

The Realtime Graphics Language romp provides a complete 3D Solids Wireframe animation system with 52 star commands and 3D Editors for designing objects to animate from your own programs. Includes a 35,000 pixels/sec line generator for fast 3D drawing rates. 3D Rotate, Scale, Orbit, Perspective and Turtlegraphics. Also compatible with designs created with SolidCAD (BBC).

£49.95 (BBC)

SUPER-DUMP

The ultimate printer driver which takes advantage of the highest resolution capability of ordinary Epson compatible printers to provide 1920 × 1024 resolution. Images can also be scaled, positioned and previewed before printing. Fully compatible with SolidCAD, Realtime Graphics Language, Gate Array design system & 3D CAD. Animation system. Your own graphics programs or other CAD packages can be made compatible with Super-Dump by the addition of a few simple commands. An example program is included in the package.

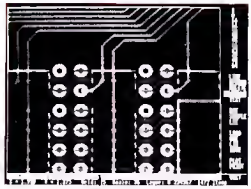
£15.95 (BBC), £24.95 (ARC) New

Presentation Manager

The package provides an interactive environment to create, edit and play-back computer presentations for lectures and demonstrations. Also handles graph plotting for polynomials & user-defined functions which can be incorporated within the presentation.

£34.95 (BBC), £49.95 (ARC) New

ARC-PCB



The ultimate PCB design system developed specifically for the Archimedes with a specification that cannot be matched. Includes Automatic routing, Rats-nesting, 8 layers, Surface mount capability, 0.001" resolution, 32" × 32" maximum board size, On-line Help, Fast Zoom, Pan/Redraw, Text & Silkscreen facility, Variable Line Pad, Text, Grid sizes, Part Libraries, Block Move/Copy/Rotate/Mirror/Erase options, and up to 300,000 components. For hardcopy, the system supports a large number of plotters and ordinary Epson compatible printers at very high resolutions (1920 × 1024) of 240 dots/inch for near laser quality output. An entry-level ARC-PCB system (without auto-routing) is available for £99.95 (Auto routing upgrade: £100). Enquire about our turnkey PCB design systems complete with Colour Archimedes and/or plotters with prices from £1220 to £3909 (Inc VAT).

£195.00 (ARC) New

RiscBASIC

Supercharge your Archimedes Basic programs by compiling them automatically into pure ARM Risc code with the RiscBASIC compiler. Features include Relocatable modules, Cross reference of all variables, functions, and procedures. Floating point and Integer support, Stand-alone code generator. Optimising compiler & Full runtime error handler.

£99.95 (ARC) New

RiscFORTH

A new 32-bit implementation of the FORTH 83 standard, designed to take full advantage of the ARM architecture. Features include Multi-tasking, Optimising compiler, built in ARM assembler with floating point mnemonics, built in Full screen Editor, File system interface, OS calls support, Floating point & Integer maths, WIMP support, Single-step debugger, Shadow screen for documentation, Block manipulation, Dictionary & Vocabulary display, Call-finding and a standalone code generator.

£99.95 (ARC) New

SILICON VISION LTD, SIGNAL HOUSE, LYON ROAD, HARROW MIDDLESEX HA1 2AG. TEL: 01-422 2274 or 01-861 2173
FAX: 01-427 5169. TELEX: 918266 SIGNAL G.
(Access/Mastercard/Eurocard accepted)
All prices include VAT and Carriage (Overseas orders add £4).

Postbag

MULTI-TASKING BASIC

I have read elsewhere that multi-tasking programs running under RISC OS cannot be written in Basic - instead they must be written in C or assembler. Surely this is very limiting? I have also heard that some of the Welcome disc utilities are written in Basic. Can you please explain this apparent anomaly.

Paul Roberts

Mr. Roberts is not alone in being confused over this matter. What he has read is that Basic cannot multi-task, i.e. that it is not possible to enter, edit, and test a Basic program in one window, while, say, using a word processor in another. But a multi-tasking application can be written in Basic, and as you say, some of the Welcome disc programs are. The RISC User Notepad elsewhere in this issue shows how this can be done.

CONFIGURING MEMORY ALLOCATIONS

I heartily condone the thoughts of Barbara Wilson (Postbag, Volume 2 Issue 2). For people like us, manuals are a little short of baffling. "Bad DIM at line 440" it said, as I tried to access the Desktop; "No room in RMA" is another of its tricks, or "No room" which is even worse. No room where? I consult the manuals - no mention of these barbaric statements, least of all how to remedy them. What is RMA anyway, and Screensize? Where is the information about the way to control them? The CONFIG. command is an unexplained maze to me.

Laura Blackburn

*To operate both effectively and efficiently, the Archimedes allocates areas of memory to different tasks at different times. Changing the memory allocation normally requires a complete reset (this is why Ctrl-Break has always been required in the past following the use of *CONFIG.).*

*RMA refers to memory allocated for relocatable modules, and much applications software is supplied in this form. The simplest solution to the RMA size problem is to QUIT from Basic (type *QUIT), and then load any relocatable module, before returning to Basic (*BA.). At this point, the Operating System can dynamically allocate memory as required.*

*Alternatively, sufficient memory can be allocated in advance using *CONFIG. Remember that on a 300 series memory is allocated for most purposes in pages of 8K, and under Arthur it is the number of pages which is defined. Thus to specify a screen size of 80K (sufficient for mode 12 for example), you should type:*

```
*CONFIG. Screensize 10
```

*and press Ctrl-Break, to allocate ten 8K pages. Under RISC OS, memory allocations can be specified in pages or 'K', and are shown as 'K' in a *STATUS display. Further, a Task Manager allows the mouse to alter dynamically a visual display of memory allocations on screen. See RISC User Volume 1 Issue 2 for more information on reconfiguring your Archimedes.*

COPYING FILES

I am having great fun with my Archimedes 310, with a 90% success rate for programs typed in. However, there's one problem I can't overcome, the transfer of data between discs in different drives. I have read the section on file transfer in the User Guide, but I still can't sort it out.

G.Drywood

*The most direct way to transfer an individual file is by using *COPY. For example, to copy file PROG1 in the current directory on drive 0 to a directory $.MyProgs on drive 1 just type:*

```
*COPY PROG1 :1.MyProgs.PROG1
```

The entire content of a directory can be copied in the same way. Thus:

```
*COPY $.Letters :1.Pipedream.Letters R
```

*will copy the directory Letters on the current drive to be a sub-directory of Pipedream on drive 1. The 'R' at the end of the command stands for 'Recurse', and is what causes the *COPY command to repeat the process with all the files in the specified directory.*

Under the RISC OS Desktop, the task is simplicity itself. Use the mouse to open and display the two directories. Dragging an image of a file from one to the other will produce a copy of the relevant file. Alternatively, select the directories or files to be copied, click on the directory to which they are to be copied, and then press the menu button to reveal a short menu from which COPY can be selected. The RISC OS Desktop really does make this and many other tasks very simple.

RU

Rounded up by Lee Calcraft

CLOSING FILES

by Sheridan Williams

In routines such as error handlers it is often important to close particular data files which have been left open. The command *CLOSE or CLOSE# 0 may be used for this purpose, but these blanket commands close *all* open files, including obey files, and others which may need to be kept open.

The alternative is to close files specifically, but the problem here is that if you attempt to close a file that is not open, errors can occur. The solution is to use the procedure PROCclosefile below. This takes as a parameter the file handle of the file to be closed, and only attempts to close the file if it is open. When it has closed the file, it resets the file handle to zero to indicate that it is closed.

```
DEFPROCclosefile(RETURN handle%)
IF handle%>0 THEN CLOSE# handle%:handl
e%=0
ENDPROC
```

RISC OS MODULE HELP

by Dennis Weaver

Although the Arthur operating system and accompanying modules give useful syntax information when you type:

```
*HELP <command>
```

the machine is less communicative when you request help on the module name itself. RISC OS is decidedly more helpful. Issuing:

```
*HELP <module name>
```

now gives information on commands and configuration options provided by the module.

SCREEN MODE AND THE RISC OS DESKTOP

by Lee Calcraft

RISC OS allows programs to be booted using Shift-Break whether the Desktop is active or not. All you need to do is to set *OPT4,3. But you may have problems with screen RAM when booting from the Desktop. Even if you have configured 160K of screen RAM, the Desktop will dynamically alter the total RAM available for the screen on the basis of the configuration of a new parameter: WimpMode. To permit a program which boots from the Desktop to use up to 160K of screen RAM, use:

```
*Configure WimpMode 15
```

followed by Ctrl-Break. This will work *regardless* of the amount of screen RAM configured. In other words, you can have a 160K screen when the screen size is configured to only 80K - all thanks to RISC OS dynamic RAM allocation.

PASSING PARAMETERS

by David Graham

The SYS call "OS_GetEnv" can be used to read any parameters supplied when a program is run. For example, the following short program stores in *name$* the full command supplied to the operating system when the program is run using *filename:

```
10 SYS "OS_GetEnv" TO name$
20 PRINT name$
```

If the program is saved under the name *test*, then typing:

```
*test fred
```

will set the variable name$ to:

```
BASIC -quit "test" fred
```

The parameter *fred* at the tail end of the string can be picked up and used in whatever way the programmer wishes. For example, as a data filename. Since the call works by picking up the last issued star command, you must ensure that no star commands are issued between running the program using *filename, and issuing the SYS call.

CARE WITH SHIFT

by Lee Calcraft

The use of Basic's shift and rotate operators ("<<", ">>" and ">>>") can give rise to syntax errors in certain circumstances because the expression evaluator appears to confuse them with the "greater than" or "less than" operators (">" or "<"). The following statement, for example, fails:

```
IF x>>>2=4 THEN VDU7
```

To resolve the problem, just bracket the expression, thus:

```
IF (x>>>2)=4 THEN VDU7
```

WHICH FILING SYSTEM?

by David Graham

It is sometimes useful in a program to be able to check which is the currently active filing system. This is easily achieved with the following SYS call:

```
SYS "OS_Args" TO fsyst
```

After making the call, the variable *fsyst* holds the number of the current filing system. The three most commonly encountered filing system numbers are:

Econet	5
ADFS	8
RAM	23

NO MORE LINE NUMBERS

by Roger Wilson (Acorn)

When Basic is called on exit from Twin, either when Twin is used without line numbers, or when line numbers have been put out of sequence in Twin, Basic automatically renumbers the incoming program with the equivalent of:

```
RENUMBER 10,10
```

A problem arises with Basic programs of longer than 6527 lines, because when numbered by tens, they will exceed the highest line number that Basic can cope with (65279).

The solution is to make a patch in RAM Basic so that it renumbers in steps of one rather than 10. To do this, find the code right at the start of Basic where the message "Program renumbered" resides, and alter the two instructions:

```
MOV R4,#10
MOV R5,#10
```

to

```
MOV R4,#1
MOV R5,#1
```

WIMP GET ICON ERROR

by David Spencer

The Programmer's Reference Manual (page 472) erroneously gives the name of the SWI call "Wimp_GetIconState" as "WimpGetIconInfo".

UTILITY BUG

by Michael Spalding

There appears to be a major bug in Arthur 1.20 (but not RISC OS) which affects the writing of utility commands - that is machine code routines with a filetype of &FFC. When a utility is invoked, the operating system allocates 1K of workspace to it, setting register R12 to point to the bottom of this block, and R13 to the top. If you use R13 as a stack pointer, then the stack grows down from the top of the workspace. However, opening a file using SWI "OS_Find" corrupts the top few bytes of the workspace, overwriting the stack and causing havoc. This was found by chance, and similar nasty behaviour might happen with other SWI calls. The cure is to move the stack lower down (using perhaps SUB R!3,R13,#256) at the start of the utility. See the Basic program lister elsewhere in this RISC User for an example of this.

UP AND DOWN BY THE PAGE

James Lovejoy

The Page Up and Page Down keys on the Archimedes are the same as the Cursor Up and Down keys, but with the effect of shift reversed. For example, if the commands:

```
*FX 4 2
*FX 225 &80
*FX 226 &90
```

were issued, then the Up and Down Cursor keys would produce ASCII codes &8F and &8E respectively, while Page Up and Page Down would generate codes &9F and &9E. However, if the Shift key was pressed, the codes would be reversed - the cursor keys producing &9E and &9F, and the Page Up and Down keys &8E and &8F. One effect of this is that almost all software that uses Shift in conjunction with Cursor Up and Down will also accept Page Up and Down as an alternative.

DELAYED SYS

David Spencer

The Archimedes Sound Scheduler module provides a call to execute a SYS command after a given delay. It takes the form:

```
SYS "Sound_QSchedule",delay,
&F000000+sys,R0,R1
```

Where *sys* is the number of the SYS call to make (found from the *Programmer's Reference Manual*), and *R0* and *R1* are the parameters to pass to it. For example:

```
$&9000="Your Time Is Up!!"+CHR$0
SYS "Sound_QSchedule",1000,&F000002,
&9000,0
```

This uses SYS call 2 which prints a text string from a given address in memory. The length of the time delay depends on the value given in the command and the TEMPO setting. For the default value of TEMPO (4096) the delay is in 1/100s of a second, so the above text is printed after 10 seconds. It is possible to queue several calls simultaneously, and this is a simple alternative to using events or interrupts. RU

BEEBUG

Your Chance To Join The Team

In 1982 when Acorn announced the release of the BBC Micro, BEEBUG was formed to provide a magazine and user group for the new computer. Since then BEEBUG has continued to grow and provide new services.

We are shortly expanding into larger premises and therefore have some interesting staff vacancies.

BEEBUG is well known not only as the publisher of RISC User, the leading magazine exclusively for the Archimedes, but as the publisher of BEEBUG, established seven years ago, and now the specialist magazine for users of the BBC Micro, Master and Compact. We also have a large showroom and Mail Order operation, supplying goods throughout the world all backed up by our Technical Support Department. Beebugsoft, our software division, provide an exciting range of products for the BBC, Master and Archimedes.

ENGINEER
Full Time & Freelance

A full time engineer is now needed to enable us to improve our repair and upgrade services. In addition to this, the successful applicant will be involved in the design and development of new hardware projects for the Archimedes, and subsequent new machines that Acorn will be releasing. We also require the services of freelance engineers to work on design and development projects from their own homes.

TECHNICAL SUPPORT ASSISTANT

Our Technical Support team provide answers and advice on an extremely wide range of subjects, covering both software and hardware on the BBC Micro, Master and Archimedes computers. We are looking for someone who owns a computer themselves and can demonstrate their enthusiasm on the subject. Training will be given, although a certain level of knowledge concerning existing products will be necessary along with the ability to express yourself clearly in writing and on the telephone. Experience on PC compatibles would also be a benefit.

If any of these positions are of interest and you would like to work in a friendly but demanding company, please apply in confidence giving full details of yourself and any relevant experience.

Personnel Manager,
BEEBUG Limited,
Dolphin Place, Holywell Hill, St. Albans, Herts.
AL1 1EX

SOFTWARE PROJECT CO-ORDINATOR

This is a new and particularly interesting position for someone who has an eye for fine details. Reporting to the software manager, the successful applicant will have the opportunity to control new software projects for the Archimedes. This is not a programming position, but knowledge of Basic and Assembler is essential (and C preferable) together with familiarity of popular applications software. The ability to write clearly will also be necessary to produce program specifications and user documentation.

RISC User Magazine Disc

April 1989

THE RISC USER NOTEPAD A multi-tasking notepad which runs under the RISC OS Desktop. This month's program includes just the basic window operations.

PROCEDURE LIBRARY
The first two procedures from the RISC User Procedure Library. The routines read mouse button and key presses, and check the mouse's position.

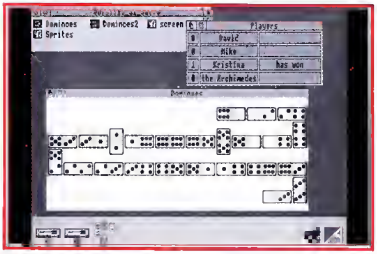
BASIC PROGRAM LISTER
A utiliy to list a Basic program directly from a file (with versions written both in Basic and machine code).

THE BASIC TOKEN TABLE
A demonstration program which lists all the Basic keywords, together with their tokens.

ARCHIMEDES VISUALS
Two image wrapping programs which run in real time.

TOOLBOX WITH RISC OS
A complete version of the RISC User Toolbox which is fully RISC OS compatible.

ACORN'S RAY TRACER
A program to ray trace objects and create animation sequences.



* BONUS ITEMS *

DOMINOES
A multi-tasking desktop dominoes game.

ART NOUVEAU
A sample screen created using this new drawing package.

COMPUTER CONCEPT'S SCANNER
An image produced with the *Scan-Light* scanner, reviewed in this issue.

THE MOLECULE
The version of the ray tracing program which draws the now famous molecule.

ARCSCAN DATA
Index entries for this issue of RISC User and the latest BEEBUG (Vol.7 No.9) to be used with *Arcscan*.

RISC User magazine discs are available to order, or by subscription. Full details of prices etc. are given on the back cover of each issue of RISC User.

RISC User BINDERS

£3.90 + £1 p&p

BEEBUG Ltd
Dolpin Place, Holywell Hill, St. Albans, Herts
AL1 1EX
Tel. (0727) 40303

RISC USER magazine

MEMBERSHIP

RISC User is available only on subscription at the rates shown below. Full subscribers to RISC User may also take out a reduced rate subscription to BEEBUG (the magazine for the BBC micro and Master series).

All subscriptions, including overseas, should be in pounds sterling. We will also accept payment by Connect, Access and Visa, and official UK orders are welcome.

RISC USER SUBSCRIPTION RATES		RISC USER & BEEBUG
£14.50	1 year (10 issues) UK, BFPO, Ch.I	£23.00
£20.00	Rest of Europe & Eire	£33.00
£25.00	Middle East	£40.00
£27.00	Americas & Africa	£44.00
£29.00	Elsewhere	£48.00

BACK ISSUES

We intend to maintain stocks of back issues. New subscribers can therefore obtain earlier copies to provide a complete set from Vol.1 Issue 1. Back issues cost £1.20 each. You should also include postage as shown:

Destination	UK, BFPO, Ch.Is	Europe plus Eire	Elsewhere
First issue	60p	£1	£2
Each subsequent issue	30p	50p	£1

MAGAZINE DISC

The programs from each issue of RISC User are available on a monthly 3.5" disc. This will be available to order, or you may take out a subscription to ensure that the disc arrives at the same time as the magazine. The first issue (with six programs and animated graphics demo) is at the special low price of £3.75. The disc for each issue contains all the programs from the magazine, together with a number of additional items by way of demonstration, all at the standard rate of £4.75.

MAGAZINE DISC PRICES	UK	Overseas
Single issue discs	£ 4.75	£ 4.75
Six months subscription	£25.50	£30.00
Twelve months subscription	£50.00	£56.00

Disc subscriptions include postage, but you should edd 60p per disc for individual orders (30p for additional discs on the same order).

All orders, subscriptions and other correspondence should be addressed to:

RISC User, Dolphin Place, Holywell Hill, St Albans, Herts AL1 1EX.
Telephone: St Albans (0727) 40303
(24hrs answerphone service for payment by Connect, Access or Visa card)